European Journal of Operational Research 326 (2025) 374–388

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Innovative Applications of O.R.

A prescriptive tree-based model for energy-efficient room scheduling: Considering uncertainty in energy generation and consumption

Siping Chen [a], Raymond Chiong [a,b], Debiao Li [c,*]

[a] School of Information and Physical Sciences, The University of Newcastle, University Drive, Callaghan, 2308, New South Wales, Australia
[b] School of Science and Technology, The University of New England, 211 Church Street, Parramatta, 2150, New South Wales, Australia
[c] Department of Management Science and Engineering, Fuzhou University, No. 2 Xue Yuan Road, Fuzhou, 350108, Fujian, China

ARTICLE INFO

Keywords:
Prescriptive analytics
Data-driven decision-making
Energy-efficient room scheduling
Uncertainty
Tree-based machine learning models

ABSTRACT

This paper investigates the energy-efficient room scheduling (ERS) problem by considering uncertainties in energy consumption and renewable energy generation in buildings. Rather than the conventional 'predict, then optimise' approach, we propose an improved prescriptive tree-based (IPTB) model that directly 'prescribes' scheduling solutions. Our model utilises contextual information on energy consumption (e.g., temperature and humidity) and renewable energies (e.g., wind speeds and sunlight) to generate direct ERS solutions. It is trained using a novel optimisation loss function that aligns historical ERS solutions with current conditions, ensuring robustness and tractability by exploiting problem-specific properties. To evaluate the proposed model's performance, experiments on randomly generated ERS instances demonstrate that the IPTB model is trained efficiently across various problem sizes and consistently outperforms advanced data-driven optimisation methods in prescriptive accuracy. Moreover, the IPTB model achieves more balanced energy consumption, particularly under practical scenarios emphasising on energy demand charges. A case study using real-world datasets from six buildings at Monash University, Australia, validates the model's effectiveness in addressing complex practical constraints inherent in ERS problems.

1. Introduction

Rising energy demands highlight the urgent need for novel solutions and innovative strategies to ensure sustainability and energy efficiency. Among the various solutions, energy conservation in buildings has emerged as a critical focus. Globally, buildings consume approximately 151 EJ of energy, which accounts for 36% of the world's final energy consumption and 25% of the carbon emissions (González-Torres et al., 2022; Santamouris & Vasilakopoulou, 2021). In Australia, buildings contributed to 24% of the nation's electricity consumption and 10% of greenhouse gas emissions in 2022 (Department of Climate Change, Energy, the Environment, and Water, 2022). To mitigate these impacts, the Australian Government anticipates a 75% increase in the deployment of renewable energy devices in buildings over the next five years. Efficiently scheduling energy activities and managing renewable energy systems offer a systematic approach to reducing carbon emissions (Fadzli Haniff et al., 2013). Accordingly, advanced scheduling algorithms have been developed to optimise energy-efficient timetabling (Sethanan et al., 2014; Song et al., 2017).

However, energy consumption and generation in buildings are often influenced by complex and interdependent uncertainties. For example, temperature fluctuations impact the energy consumption of air conditioning systems, while unpredictable cloud cover affects the sunlight received by solar panels. These uncertainties compromise the reliability of the energy-aware scheduling solutions, frequently resulting in increased energy consumption of building equipment and lower utilisation of renewable devices. While uncertainties and the corresponding forecasting methodologies are explored in the literature (Dudek et al., 2021; Hu et al., 2019; Khodayar et al., 2020), their implications for energy-aware decision-making remain under-examined.

The predict+optimise (PO) is a general approach for solving optimisation problems under uncertainty and has been applied to energy-aware decision-making tasks (Hu et al., 2019; Li & Chiang, 2016). However, it primarily employs machine learning (ML) models to make prediction of uncertainty, neglecting other critical aspects of decision-making, such as feasibility and optimality. Even minor deviations in predictions can result in contradictory decisions and substantial decision losses (Pogancic et al., 2020). To overcome these limitations, integrated learning and optimisation (ILO) approaches have emerged as promising tools that combine ML with optimisation techniques to

* Corresponding author.
E-mail address: debiaoli@fzu.edu.cn (D. Li).

https://doi.org/10.1016/j.ejor.2025.02.023
Received 2 September 2023; Accepted 18 February 2025
Available online 28 February 2025

address such challenges. While predictive models focus on forecasting future events from contextual data, Bertsimas and Kallus (2019) presented prescriptive models that take one step further by directly 'prescribing' the optimised decision to the optimisation problem given the same contextual data. Similarly, Elmachtoub and Grigas (2021) proposed a novel framework called smart "predict, then optimise" (SPO), which uses the optimisation loss function as the ML training objective. However, existing methods either rely on surrogate loss functions (Elmachtoub & Grigas, 2021), which have strong linear conditions for asymptotic optimality, or focus on problem-specific training approaches, such as those of Kallus and Mao (2023). In more complex decision-making environments, uncertainties can significantly impact scheduling feasibility, with the problem itself often being NP-hard. These structural complexities create conflicts with the asymptotic optimality conditions of current methods, thereby compromising the performance of trained SPO or prescriptive models.

In this paper, we address an energy-efficient room scheduling (ERS) problem, which accounts for uncertainties in building energy consumption and renewable energy generation. This problem falls under the category of resource-constrained project scheduling problems and is applicable to other energy-aware applications in similar stochastic environments. The presence of time-series uncertainties poses significant challenges in defining feasible domains, complicating the application of traditional methods. To overcome these challenges, we propose an improved prescriptive tree-based (IPTB) model with enhanced prescriptive capabilities. The model's performance is evaluated through experiments using several ML-integrated data-driven optimisation (DDO) methods on randomly generated datasets. Comparative results demonstrate that the proposed model achieves superior runtime efficiency and accuracy compared to existing optimisation tree loss functions. Additionally, a case study on a real-world ERS problem, leveraging datasets from the IEEE-CIS technical challenge (Bergmeir, 2021), validates the practical effectiveness of our approach.

This study's contributions are summarised as follows:

- This paper is the first study to present a prescriptive tree-based (PTB) approach for a contextual and stochastic ERS problem. Our method leverages reformed solution data for tree-splitting, enabling the generation of feasible scheduling solutions.
- We propose a novel loss function, a downstream optimisation loss computed by the proposed decision-prescribing formulation, leading to further improvements in solution quality and avoiding repetitively computing the optimisation problems during training.
- Our prescriptive approach mitigates the optimisation loss induced by prediction and outperforms other well-established PO and ILO methods.
- The case study demonstrates that embedding the proposed approach within a rolling-horizon ERS process dynamically adapts to emerging uncertainty patterns, enhancing robustness and reliability.

The rest of this paper is organised as follows. Section 2 reviews the relevant literature, while Section 3 presents the mathematical model and PO framework for the ERS problem. In Section 4, we describe the proposed IPTB model in detail. In Section 5, we discuss our experimental results and analysis. Finally, we provide our conclusions and suggestions for future work in Section 6.

2. Related work

In this section, we first review the literature related to the ERS problem. This is followed by a discussion of energy-related forecasting studies. Given the limited research on ERS problems that incorporates DDO tools, we discuss the connection between our study and different DDO methods.

2.1. Energy-efficient room scheduling

The room scheduling problem aims to organise activities by timetabling them into rooms that meet the needs of occupants. Educational buildings are a common focus of research on room scheduling problems, and different timetabling algorithms have been developed to assign lectures to classrooms according to specific lecture requirements (Akbarzadeh & Maenhout, 2021; Bagger et al., 2019; Cacchiani et al., 2013; Esmaeilbeigi et al., 2022). Further, Cardoen et al. (2010) highlighted that operating room scheduling in hospitals and clinics has received significant research interest, given that each surgery requires a combination of human resources, equipment, and materials (Shao et al., 2023). These studies demonstrate that various optimisation objectives need to be considered in room scheduling problems. This includes energy-related objectives, which have been an important research topic in the scheduling field (Abedi et al., 2024; Zhang & Chiong, 2016; Zhang et al., 2023).

Most research on ERS focuses on reducing the energy consumption of heating, ventilation, and air conditioning (HVAC) systems. Major attempts focus on scheduling HVAC systems themselves to improve energy efficiency (Escrivá-Escrivá et al., 2010; Jiang et al., 2021). Yang and Becerik-Gerber (2014) suggested an approach for scheduling HVAC systems to improve energy efficiency. However, the performance of HVAC schedules can be significantly decreased without a proper activity schedule. To address this, many studies proposed various algorithms for different scheduling scenarios. For example, Sethanan et al. (2014) proposed a classroom scheduling algorithm to reduce energy use in classrooms. Similarly, Song et al. (2017) introduced an energy efficiency-based course timetabling algorithm to identify optimal timetables in terms of energy usage.

As previously discussed, the existing literature demonstrates that appropriate room allocations can significantly reduce energy consumption and enhance occupants' satisfaction. However, how uncertainties affect room scheduling, especially regarding energy efficiency, has not been studied much. Moreover, although some research has combined energy-related forecasting tasks with energy-efficient decision-making procedures (Li & Chiang, 2016), the potential losses arising from the transition between predictive models and downstream optimisation tasks are hardly discussed in this field. This issue can be critical when applying optimisation models, as highlighted by Elmachtoub and Grigas (2021). Limited attempts have been made to fill this gap by integrating the forecasting and scheduling phases in ERS problems.

2.2. Energy-related forecasting and data-driven optimisation

Various energy-related uncertainties can be found in energy-efficient scheduling problems, such as device energy consumption and renewable energy generation. Much of the existing research has focused on forecasting energy production and consumption, including energy demand (Dudek et al., 2021), renewable power generation (Khodayar et al., 2020), and energy pricing (Hu et al., 2019). Additionally, an increasing number of studies have explored ML methods that combine prediction and optimisation by using customised loss functions (Li & Chiang, 2016). Another study employed meta-optimisation for energy-related forecasting, which involved incorporating an additional optimisation layer to the PO pipeline (Carriere & Kariniotakis, 2019). These studies addressed the importance of applying the PO framework in energy-related optimisations that involve uncertainties.

In the ILO framework, forecasting and optimisation are not considered separate phases. Instead, their interaction is more extensively addressed within the ILO paradigm (Sadana et al., 2024). With this approach, rather than relying on different measures of forecast quality such as the mean absolute error, forecasting tasks are evaluated according to their contribution to the actual downstream cost of the optimisation problem (Kotary et al., 2021). For example, a dynamic-programming–based PO framework was proposed by Demirovi et al.

Table 1
Summary of ILO studies. (Abbreviations for this table—obj.: objective function; const.: constraint space; LP: Linear Programming; ILP: Integer Linear Programming).

Study	Training	Problem	Method	Uncertainty
Elmachtoub et al. (2020)	SPO loss	LP	PO	In obj.
Tian et al. (2023)	SPO loss	ILP	PO	In obj.
Mandi et al. (2020)	SPO loss	ILP (NP-hard)	PO	In obj.
Yan et al. (2020)	SPO loss	ILP	PO	In obj.
Bertsimas and Kallus (2019)	Prediction loss	ILP	wSAA	In obj.
Kallus and Mao (2023)	Oracle-splitting	ILP	wSAA	In obj.
Stratigakos et al. (2022)	Optimisation loss	LP	wSAA	In obj.
This work	Optimisation loss	ILP (NP-hard)	wSAA	In obj. and const.

(2020) to minimise the regret in stochastic optimisation problems. Elmachtoub et al. (2020) developed a specialised algorithm to build decision trees directly for the actual optimisation target. In a separate study, Elmachtoub and Grigas (2021) introduced a differentiable surrogate for the SPO loss function. Building on this research stream, Mandi et al. (2020) applied these concepts to real-world optimisation problems and proposed techniques for efficiently learning the model. Some extensive research was also conducted by Jeong et al. (2022) to develop a symbolic technique-based SPO framework that solves the optimisation problem optimally. Another notable contribution comes from Bertsimas and Kallus (2019), who proposed a prescriptive optimisation method that combines weighted sample average approximation (wSAA) with an ML model. Kallus and Mao (2023) integrated the optimisation loss into a prescriptive model and applied an oracle splitting criterion for prescriptive tree construction. These methods have been applied to decision-making problems, such as cargo inspection planning (Yan et al., 2020), cost-sensitive multi-class classification (Jeong et al., 2022), and entropy-constrained portfolio allocation (Liu & Grigas, 2021). In the context of energy-related optimisation, only a study conducted by Stratigakos et al. (2022) applied prescriptive trees to an energy trading problem, which does not directly serve as a relevant example for the planning and scheduling problems addressed in this paper.

We present these existing ILO methods, their applied methodology and problem characteristics in Table 1. All these existing methods exhibit limitations when applied to an ERS problem. For example, SPO-loss-based methods require the optimisation problem to have a strictly linear objective function, which means that uncertainties should not affect the constraint space. Similarly, current prescriptive methods either rely on prediction loss or impose the same restrictions on the constraint space when learning the weight function for wSAA (Stratigakos et al., 2022). In the context of scheduling, uncertainties typically impact the constraint space, creating a conflict with the assumptions of these methods. Practical examples include uncertain surgery durations affecting operating room schedules (Khaniyev et al., 2020) and random machine failures impacting production schedules (Lu et al., 2015). This also decreases the ML model's performance when applying existing methodologies. Considering the problem's complexity, all existing methods using optimisation loss can deal with a relatively simple optimisation problem. However, we emphasise the significant computational burden associated with calculating optimisation losses for practical scheduling problems. Although Mandi et al. (2020) proposed to use a weak oracle and initialisation to accelerate the training, it is still time-consuming due to the need to repetitively calling the optimisation solver. Further research is therefore necessary.

3. Problem description

We formulate the ERS problem as an activity timetabling problem—more specifically, as a resource-constrained project scheduling problem with time windows (Bartusch et al., 1988). In this problem, an activity schedule is generated over a planning horizon, which refers to the period for making all scheduling decisions. The planning horizon is divided into $|T|$ time slots of fixed intervals. For example, a weekly schedule segmented into 15-minute intervals has $|T| = 672$ time slots. Activities in the activity set A must be scheduled into one or more rooms appropriately. Each activity $a \in A$ can begin within its feasible starting time slots $\mathcal{S}_a$ and progress for a duration σ_a without preemption. Each activity also occupies a different number of large rooms r^l_a or small rooms r^s_a. Additionally, some activities are subject to precedence constraints, which ensure that activity a is scheduled one or more days after its preceding activities, as defined by its precedent activity data $\mathcal{D}_a$. This reflects common scheduling practices such as ensuring that a lab session should not be scheduled on the same day or before the corresponding course. The original resource-constrained project scheduling problem has been proven to be NP-hard (Bartusch et al., 1988), and the ERS problem, an extension of this original problem with an additional precedence constraint, is also NP-hard. This paper assumes at least one feasible ERS solution exists in which all activities can be arranged. To ensure this assumption, all instances in this paper are generated based on feasible ERS solutions. Related notation is listed in Table 2.

In this paper, each building has uncertain baseline energy consumption, while each renewable energy device (e.g., solar panel array) has uncertain energy generation. We model m sources of uncertainty, accounting for the non-identical distributions of building energy consumption, driven by variations in size, function, and equipment. Similarly, energy generation from renewable devices is non-identically distributed due to contextual and environmental factors. If the full probability distributions of these uncertainties were known, the problem could be directly addressed. However, obtaining these distributions is impractical in real-world applications. Instead, decision-makers observe historical uncertainty values and contextual features influencing energy consumption (e.g., temperature, humidity) and energy generation (e.g., sunlight, thermal radiation). To overcome these challenges, we reformulate the stochastic ERS problem within a PO framework. This data-driven, distribution-free optimisation approach relies solely on observed uncertainty data and contextual features, eliminating the need for explicit knowledge of uncertainty distributions. The n historical observations of m uncertainties and q contextual features are represented as the following matrices:

$$\begin{pmatrix} c_{11} & \dots & c_{1m} \\ \dots & \dots & \dots \\ c_{n1} & \dots & c_{nm} \end{pmatrix} \begin{pmatrix} x_{11} & \dots & x_{1q} \\ \dots & \dots & \dots \\ x_{n1} & \dots & x_{nq} \end{pmatrix}$$

Each row in these matrices corresponds to an observation collected from one of $N = n/|T|$ historical scenarios, where c_{tk} represents the uncertainty value for the kth uncertainty at time slot t, and x_{tj} denotes the jth contextual feature at the same time slot. The predictive model for kth uncertainty is trained by minimising prediction loss function l_p (e.g., mean absolute loss function or mean square loss function). With all trained models, the PO framework conducts point estimations of all the uncertain energy consumptions and generations for $t \in T = \{n+1, \dots, n+|T|\}$. These predictions are subsequently combined to calculate y_t, the total base energy consumption at time slot t, by subtracting total energy generation from total energy consumption. Using these values, the ERS problem can then be solved deterministically by optimising the following problem:

Table 2
A notation list for the ERS problem.

Sets and Indices	
I	The set for N historical scenarios, $i, i' \in I = \{1, \dots, N\}, i \neq i'$
A	The set for $\|A\|$ activities, $a \in A = \{1, \dots, \|A\|\}$
T	The set for $\|T\|$ time slots for current planning horizon, $t \in T = \{n+1, \dots, n+\|T\|\}$
J	The set for q auxiliary features from $j \in J = \{1, \dots, q\}$
Parameters	
n	The number of historical time slots, $n = \|T\| * N$
x_{jt}	The observation of jth auxiliary feature in time slot t
c_{tk}	The kth uncertainty's value in time slot t
$\hat{c}_{tk}$	The prediction of kth uncertainty's value in time slot t
y_t	The base energy consumption (activity-free) in time slot t
h_a	The energy consumption of activity a
$\mathcal{S}_a$	The feasible starting time slots of activity a
$\mathcal{D}_a$	The precedent activities of activity a
p_t	The energy price in time slot t
σ_a	The duration of activity a
r_a^s	The number of small rooms occupied by activity a
r_a^l	The number of large rooms occupied by activity a
n_l	The total number of large rooms
n_s	The total number of small rooms
Data	
$\boldsymbol{x}_i$	A row of observed contextual features of ith ERS scenario
X	The observation data of N historical observations, $\boldsymbol{x}_i \in X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$
$\boldsymbol{y}_i$	The array of $\|T\|$ uncertainty values associated with the ith ERS scenario.
Y	The uncertainty data of N historical ERS scenarios, $\boldsymbol{y}_i \in Y = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_N\}$
Z	Solution data for reformed ERS scenarios, $z^*(\boldsymbol{y}_i) \in Z = \{z^*(\boldsymbol{y}_1), \dots, z^*(\boldsymbol{y}_N)\}$
U	An $N \times N$ cache storing objective values, $u_{ii'} = \mu(z^*(\boldsymbol{y}_i), \boldsymbol{y}_{i'}) \in U$
Variables	
s_{at}	The decision variable that activity a starts in time slot t
v_{at}	The decision variable that activity a processes in time slot t
d_a	The decision variable of the starting date of activity a
e_t	The total energy consumption in time slot t
$e_{\max}$	The maximum energy consumption of e_t
z	A feasible ERS solution, defined as a set of decision variable values.
$z^*(\boldsymbol{y}_i)$	The optimal deterministic ERS solution for scenario i
Functions	
$\mu(z, \boldsymbol{y}_i)$	The objective function of ERS problem given an ERS solution z and uncertainty array $\boldsymbol{y}_i$
$C(z)$	The function of computing total energy costs given an ERS solution z
$G(z, \boldsymbol{y}_i)$	The function of retrieving the maximum energy consumption ($e_{\max}$) of solution z applied in scenario $\boldsymbol{y}_i$.
$l_p(\cdot)$	The prediction loss function
$l_{\text{SPO}}(\cdot)$	The SPO tree optimisation loss function
$l_{\text{SP}}(\cdot)$	The proposed optimisation loss function
$z^{PTB}(\boldsymbol{x})$	The PTB decision given an observation array $\boldsymbol{x}$
$z^{\text{SAA}}(Y)$	The SAA decision given a uncertainty data Y
$z^{\text{SP}}(Y)$	The SP decision given uncertainty data Y
$\mathcal{L}(\cdot)$	The constraint space of ERS problem

$$\text{minimise} \quad \sum_{t \in T} (p_t \cdot e_t) + \gamma \cdot e_{\max} \tag{1}$$

$$\text{s.t.} \quad \sum_{t' \in \mathcal{T}_a} s_{at'} = 1 \quad \forall a \tag{2}$$

$$\sum_{t' \in \mathcal{S}_a \cap [t-\sigma_a+1, \dots, t]} s_{at'} = v_{at} \quad \forall a, \forall t \in \mathcal{S}_a, \tag{3}$$

$$d_{a'} + 1 \leq d_a \quad \forall a, \forall a' \in \mathcal{D}_a \tag{4}$$

$$\lceil \frac{t}{|D|} \rceil s_{at} = d_a \quad \forall a, \forall t \tag{5}$$

$$\sum_{a \in A} r_a^l \cdot v_{at} \leq n_l \quad \forall t \tag{6}$$

$$\sum_{a \in A} r_a^s \cdot v_{at} \leq n_s \quad \forall t \tag{7}$$

$$e_t = \sum_{a \in A} s_{at} \cdot h_a + y_t \quad \forall t \tag{8}$$

$$e_{\max} = \max_{t \in T}(e_t) \tag{9}$$

$$v_{at} \in \{0, 1\}, s_{at} \in \{0, 1\} \quad \forall a, \forall t \tag{10}$$

$$d_a \in \mathbf{Z}_+ \quad \forall a \in A \tag{11}$$

$$e_t \geq 0 \quad \forall t \tag{12}$$

$$e_{\max} \geq 0 \tag{13}$$

As indicated in Eq. (1), the optimisation objective function minimises two energy-related terms: total energy cost and energy demand charge. The energy cost is calculated as the sum of total energy consumption e_t multiplied by the corresponding energy price p_t across all periods: $\sum_{t \in T}(p_t \cdot e_t)$. The energy demand charge $\gamma \cdot e_{max}$ represents an additional charge incurred during the highest energy demand time slot $e_{\max} = \max_{t \in T}(e_t)$. The parameter γ is a weight that regulates the relative importance of the energy demand charge compared to the energy cost. This linear function aims to reduce overall economic energy costs while balancing energy usage (Jiang et al., 2021).

Constraint (2) ensures that each activity a starts within its feasible starting time slot set $\mathcal{S}_a$. Constraint (3) enforces decision variable $v_{at} = 1$ if activity a is still in progress during time t. Constraint (4) mandates that a successor activity a can be scheduled at the earliest after $d_{a'}$, where $d_{a'}$ represents the day on which the precedent activity $a' \in \mathcal{D}_a$ is scheduled. In Constraint (5), the total number of time slots in a day ($|D|$) is used to compute the values of d_a from the starting time decision variable s_{at}. Constraints (6) and (7) address room capacity requirements, where n_l and n_s represent the total number of large rooms and small rooms, respectively. Constraint (8) defines the equation for calculating the total energy consumption e_t in time slot t, where h_a is a parameter representing an activity's energy consumption. This study

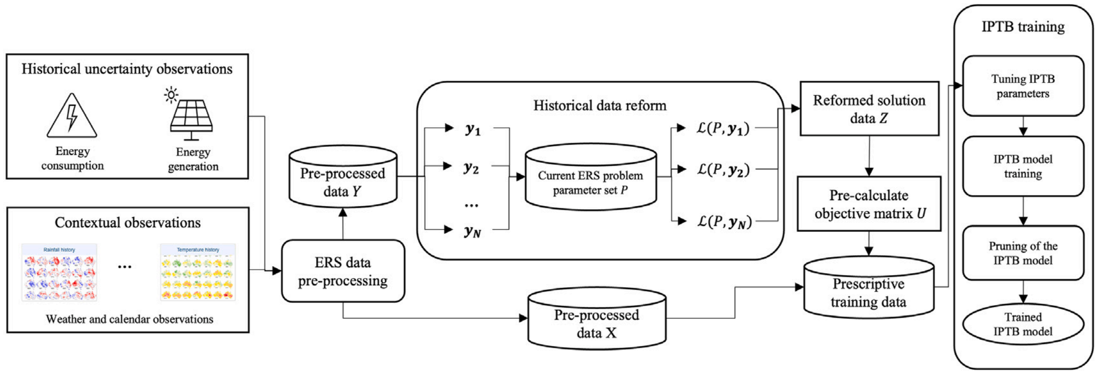


Fig. 1. A flowchart of the proposed IPTB model for the ERS problem.

assumes that contextual information does not influence activity energy consumption since HVAC consumption has been considered as our uncertainty. Instead, activity energy consumption is estimated based on the activity type and the number of attendees, with this information provided by the parameter h_a. Constraint (9) computes the maximum energy consumption $e_{\max}$ across all time slots t in T. Finally, Constraints (10)–(13) define the domains for all decision variables.

In the PO procedure, Eq. (1) is solved optimally to obtain a solution. This solution can vary depending on the uncertainty values it receives; for instance, it may differ across historical scenarios. To better understand the ERS problem in this context, we analyse it under the PO framework and propose several ERS properties. For clarity, we introduce uncertainty array $\boldsymbol{y}_i$, which comprises all y_t values for the time slots in scenario i. The optimal solution for $\boldsymbol{y}_i$, denoted by $z^*(\boldsymbol{y}_i)$, is a set of decision variable values that satisfies the constraints and minimises the objective function. Additionally, $\mu(z, \boldsymbol{y}_i)$ denotes the objective function of the deterministic ERS problem, given any feasible solution z and uncertainty array $\boldsymbol{y}_i$. The first property is expressed as follows.

Property 3.1. *$z^*(\boldsymbol{y}_i) = z^*(\boldsymbol{y}_i - b)$, where b is a real number.*

This property can be proven by considering the linearity of the objective function. According to Eq. (1), both activity energy consumption $\sum_{t\in T}(p_t \cdot e_t)$ and energy demand charge $\gamma \cdot e_{\max}$ are linear when $\boldsymbol{y}_i$ is provided. When subtracting a value b from $\boldsymbol{y}_i$, both the activity energy consumption and energy demand charge remain optimal, thus establishing $z^*(\boldsymbol{y}) = z^*(\boldsymbol{y} - b)$.

The second property concerns two different uncertainty arrays, $\boldsymbol{y}_i$ and $\boldsymbol{y}_{i'}$. We introduce $C(z)$ to represent the total activity energy cost of a feasible solution z. Note that $C(z)$ depends only on z and not on $\boldsymbol{y}_i$, as the parameter h_a (activity energy consumption) and p_t (energy price) are fixed and given. Moreover, $G(z, \boldsymbol{y}_i)$ represents the highest energy demand of z in $\boldsymbol{y}_i$, corresponding to the value of $e_{\max}$ used in $\mu(z, \boldsymbol{y}_i)$.

Property 3.2. *If $G(z^*(\boldsymbol{y}_i), \boldsymbol{y}_{i'}) = G(z^*(\boldsymbol{y}_{i'}), \boldsymbol{y}_{i'})$ and $G(z^*(\boldsymbol{y}_i), \boldsymbol{y}_i) = G(z^*(\boldsymbol{y}_{i'}), \boldsymbol{y}_i)$, activity energy costs $C(z^*(\boldsymbol{y}_i)) = C(z^*(\boldsymbol{y}_{i'}))$ and solutions $z^*(\boldsymbol{y}_i)$ and $z^*(\boldsymbol{y}_{i'})$ are interchagable and remain optimal for both $\boldsymbol{y}_i$ and $\boldsymbol{y}_{i'}$.*

Proof. Assume $C(z^*(\boldsymbol{y}_i)) > C(z^*(\boldsymbol{y}_{i'}))$. Given the condition $G(z^*(\boldsymbol{y}_i), \boldsymbol{y}_i) = G(z^*(\boldsymbol{y}_{i'}), \boldsymbol{y}_i)$, it follows that: $C(z^*(\boldsymbol{y}_i)) + \gamma G(z^*(\boldsymbol{y}_i), \boldsymbol{y}_i) > C(z^*(\boldsymbol{y}_{i'})) + \gamma G(z^*(\boldsymbol{y}_{i'}), \boldsymbol{y}_i)$. Adding the term $\sum_{t\in \boldsymbol{y}_i} p_t \cdot y_t$ to both sides of this inequality gives: $\mu(z^*(\boldsymbol{y}_i), \boldsymbol{y}_i) > \mu(z^*(\boldsymbol{y}_{i'}), \boldsymbol{y}_i)$. This contradicts the condition that $z^*(\boldsymbol{y}_i)$ is optimal for $\boldsymbol{y}_i$, which requires $\mu(z^*(\boldsymbol{y}_i), \boldsymbol{y}_i) \le \mu(z^*(\boldsymbol{y}_{i'}), \boldsymbol{y}_i)$. Consequently, the assumption $C(z^*(\boldsymbol{y}_i)) > C(z^*(\boldsymbol{y}_{i'}))$ cannot hold under the given conditions. Similarly, assume $C(z^*(\boldsymbol{y}_i)) < C(z^*(\boldsymbol{y}_{i'}))$. By applying the same reasoning, we arrive at: $\mu(z^*(\boldsymbol{y}_i), \boldsymbol{y}_{i'}) < \mu(z^*(\boldsymbol{y}_{i'}), \boldsymbol{y}_{i'})$, which again contradicts the optimality of $z^*(\boldsymbol{y}_{i'})$. Therefore, $C(z^*(\boldsymbol{y}_i)) = C(z^*(\boldsymbol{y}_{i'}))$. This also indicates that the solutions $z^*(\boldsymbol{y}_i)$ and $z^*(\boldsymbol{y}_{i'})$ are optimal for both historical scenarios i and i', completing the proof. □

This property implies that a group of ERS scenarios can be represented by other scenarios' optimal solutions.

4. Methodology

In this section, we present our IPTB model for prescriptive decision-making in the ERS problem. Previous studies have demonstrated that tree-based models, such as decision trees and random forest (RF) models, are effective for energy-related forecasting tasks (Hong et al., 2016). However, the conventional PO framework typically separates the prediction and optimisation phases, which may compromise the overall performance of the model. To overcome this limitation, the IPTB model systematically integrates prediction and optimisation into a unified framework.

Fig. 1 illustrates the IPTB model process for the ERS problem. Initially, the data from uncertain and contextual observations is pre-processed to generate datasets X and Y. A crucial step, namely historical data reform, extracts uncertainty arrays $\boldsymbol{y}_1, \dots, \boldsymbol{y}_N$ and combines them with the current ERS problem constraint space $\mathcal{L}$ to create reformed ERS scenarios. By solving these scenarios, reformed scheduling solutions and their corresponding pre-calculated objective values are obtained. Subsequently, the IPTB training algorithm is applied, using the proposed optimisation loss function as the training objective. Furthermore, additional pruning techniques are employed to improve training efficiency and prediction accuracy while mitigating the risk of overfitting.

4.1. Historical prescriptive data

In this section, we describe the data construction process that transforms predictive data into prescriptive data to facilitate IPTB training. We first propose a method to obtain reformed ERS solutions, as illustrated in Fig. 2. Given the uncertainty data, we initially calculate the total base energy consumption y_t by subtracting the total energy generation from the total energy consumption. The uncertainty data is then separated into $|T|$-sample uncertainty arrays based on their calendar features. These uncertainty arrays are combined with the current ERS parameters to create N reformed ERS scenarios. The ERS model is then applied to solve these reformed scenarios optimally, yielding reformed solution data $Z = \{z^*(\boldsymbol{y}_1), z^*(\boldsymbol{y}_2), \dots, z^*(\boldsymbol{y}_N)\}$. Subsequently, each solution $z^*(\boldsymbol{y}_i) \in Z$ is applied to other reformed instances to calculate the corresponding objective function values $u_{ii'} = \mu(z^*(\boldsymbol{y}_i), \boldsymbol{y}_{i'}), \forall \boldsymbol{y}_{i'} \in Y$. These objective values are compiled into a matrix U, which is later used to compute the optimisation loss value.

Incorporating the reformed solution data Z and the associated objective values matrix U significantly enhances the quality of prescriptive decision solutions. When a prescriptive model is trained using prediction loss metrics, such as the mean squared error (MSE) as in Bertsimas and Kallus (2019), the model selects the historical scenarios with small differences in objective values. However, in scheduling

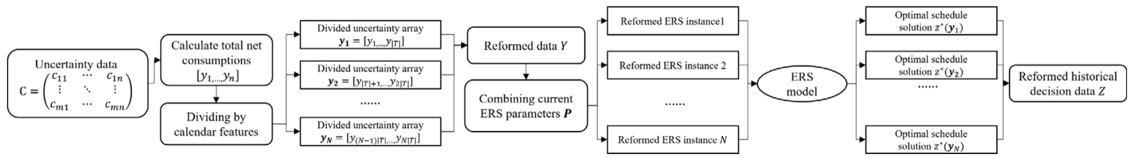


Fig. 2. The procedure of historical data reform.

problems – or other complex combinatorial optimisation problems – similar objective values can correspond to entirely different solutions. This lack of consistency between selected samples can result in substantial optimisation loss, especially when constraints are highly complex. By integrating Z and U into the training process, the focus shifts towards directly minimising the optimisation loss using solutions from Z. This targeted approach is more effective as it accounts for variations between similar objective values and leverages the diversity of possible scheduling solutions, enhancing decision quality while improving robustness against uncertainty (see Fig. 2).

We pre-process the observation data X and normalise uncertainty data Y to enhance their suitability for the ERS problem. Feature engineering procedures are conducted to reduce the input feature dimensions. For instance, some weather features are recorded hourly, resulting in high-dimensional input data. To address this, we aggregate these features into daily or weekly representations by computing their mean, median, or standard deviation values. This process preserves the underlying patterns of these features while reducing dimensionality. After dimensionality reduction, ERS-based normalisation is conducted according to Property 3.1. We use this property to normalise each total net energy consumption value y_t from an uncertainty array $\boldsymbol{y}_i \in Y$ by the expression $\frac{y_t - \min(\boldsymbol{y}_i)}{\max(\boldsymbol{y}_i)}$. According to Property 3.1, subtracting a fixed value $\min(\boldsymbol{y}_i)$ from the array of uncertainties $\boldsymbol{y}_i$ will not change the optimal solution. This standardisation reduces redundant information of input data, enabling the ML model to identify similar ERS instances more efficiently. Finally, we use the $|T|$-time-unit median value to impute missing values in the dataset. This imputation method mitigates the impact of outliers in the training data.

4.2. Improved prescriptive tree-based model

In this subsection, we detail the IPTB model and its training procedures. For PO problems, Bertsimas and Kallus (2019) proposed a PTB model that prescribes optimal decisions based solely on observation data. This model employs a decision tree-controlled wSAA procedure, which is defined by the following equation:

$$z^{\mathrm{PTB}}(\boldsymbol{x}) = \arg\min_{z\in\mathcal{L}} \sum_{i=1}^{N} w_i^{DT}(\boldsymbol{x})\mu(z, \boldsymbol{y}_i) \tag{14}$$

Here, $w_i^{DT}(\boldsymbol{x})$ represents the weight function of the ith historical ERS scenario, trained by the decision tree model. This weight function is defined as:

$$w_i^{DT}(\boldsymbol{x}) = \frac{\mathcal{I}(R(\boldsymbol{x}) = R(\boldsymbol{x}_i))}{|\{j : R(\boldsymbol{x}_j) = R(\boldsymbol{x})\}|},$$

where $\mathcal{I}(\cdot)$ is an indicator function that equals 1 if the condition inside is true and 0 otherwise; $|\cdot|$ denotes the number of elements in a set; and R represents the leaf-splitting function of the trained decision tree (Breiman et al., 2017). It is important to note that if $w_i^{DT} = \frac{1}{N}$ for all i, the z^{PTB} solution reduces to the SAA solution $z^{\mathrm{SAA}}(Y)$, which utilises the entire reformed uncertainty data Y. Fig. 3 illustrates the PTB decision-making process for the ERS problem. The contextual features are used to split the data, and each resulting subset, clustered in the leaf nodes, is then utilised to compute the solution z^{PTB}.

As shown in Eq. (14), the weight $w_i^{DT}(\boldsymbol{x})$ is used to evaluate the ith historical scenario's similarity to input data $\boldsymbol{x}$. This weight function is trained using prediction loss l_p. However, l_p can only identify similar

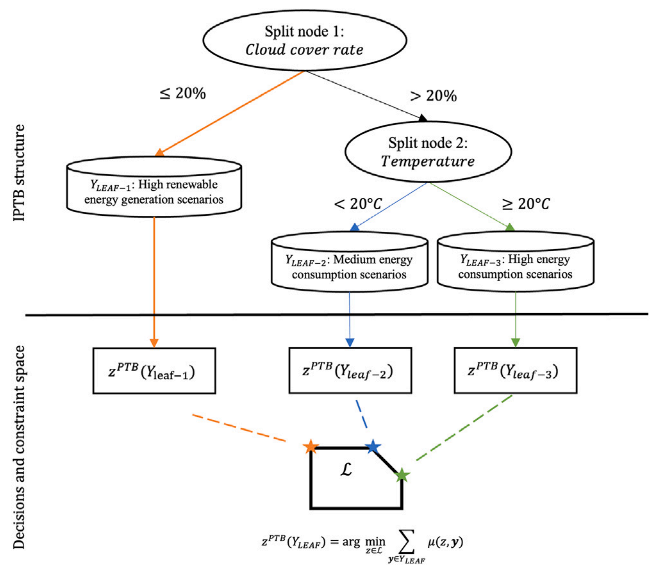


Fig. 3. The decision-making process of a PTB model on the ERS problem.

objective values instead of differentiating which schedule is more effective (Pogancic et al., 2020). This results in deviations when selecting SAA samples, which in turn reduces the model's prescriptiveness. Elmachtoub et al. (2020) proposed an optimisation objective l_{SPO} to train a tree-based model according to true optimisation objectives, which is expressed in Eq. (15):

$$l_{\mathrm{SPO}}(j,s) = \frac{1}{N}\left[\sum_{i\in R_1(j,s)} \left(\mu(z^*(\bar{\boldsymbol{y}}_l), \boldsymbol{y}_i) - u_{ii}\right) + \sum_{i\in R_2(j,s)} \left(\mu(z^*(\bar{\boldsymbol{y}}_r), \boldsymbol{y}_i) - u_{ii}\right)\right] \tag{15}$$

where $R_1(j,s)$ and $R_2(j,s)$ denote the left and right split functions in the decision tree's structure, which split the observation data on their jth feature according to their sth value; $u_{ii} = \mu(z^*(\boldsymbol{y}_i), \boldsymbol{y}_i)$; and $\bar{\boldsymbol{y}}_l$ and $\bar{\boldsymbol{y}}_r$ represent an averaged uncertainty array of the within-leaf sample set Y_l and Y_r obtained by $R_1(j,s)$ and $R_2(j,s)$, respectively. Nevertheless, this formulation is feasible only if $z^*(a\boldsymbol{y}) = z^*(\boldsymbol{y})$ (i.e., it is a strongly linear problem). In the ERS problem, this condition is violated because the energy demand charge includes a non-strictly linear max term that depends on the uncertainty y_t. Hence, combining l_{SPO} with the PTB model could not prescribe an ERS solution accurately. Considering the limitation of l_{SPO}, we first reformulate l_{SPO} by prescriptive formulation for problems where $z^*(a\boldsymbol{y}) \neq z^*(\boldsymbol{y})$, as shown in Eq. (16):

$$l_{\mathrm{SPO}}(j,s) = \frac{1}{N}\left[\sum_{i\in R_1(j,s)} (\mu(z^{\mathrm{SAA}}(Y_l), \boldsymbol{y}_i) - u_{ii}) + \sum_{i\in R_2(j,s)} (\mu(z^{\mathrm{SAA}}(Y_r), \boldsymbol{y}_i) - u_{ii})\right] \tag{16}$$

This formulation can prescribe a robust decision by identifying appropriate SAA samples for each leaf node in the decision tree. However, training with this function has two major drawbacks. On the one hand, computing this loss is time-consuming because it requires multiple computations of z^{SAA}. These computations are particularly slow during

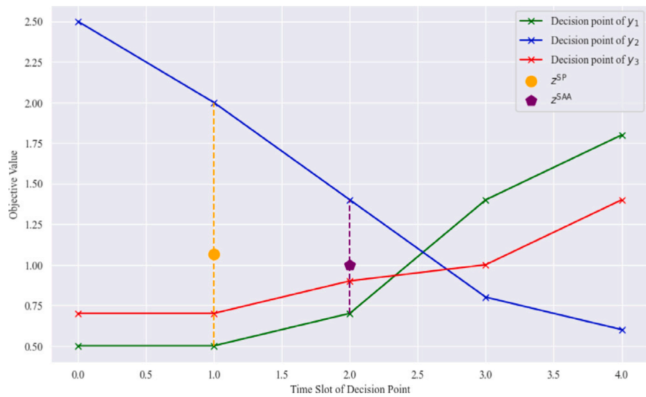


Fig. 4. Comparison of z^{SP} and z^{SAA} for a PTB example in a single-activity ERS problem. The leaf data Y_{leaf} consists of three historical scenarios $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3\}$. The heights of z^{SP} and z^{SAA} represent averaged objective values among all scenarios..

the initial layers of the decision tree, where a large number of data samples must be processed to perform SAA calculations. Second, in the ERS context, we require a single scheduling solution that maximises utility for the current planning horizon. z^{SAA} often produces averaged decisions, which can inadvertently cluster together unrelated ERS scenarios with different optimisation patterns into a single split.

To improve the efficiency and accuracy of PTB model training, we propose z^{SP} alternative decision formulation for the PTB model, as shown in Eq. (17):

$$z^{SP}(Y) = \arg\min_{z \in Z} \sum_{\boldsymbol{y}_i \in Y} \mu(z, \boldsymbol{y}_i) \tag{17}$$

Since the values of $\mu(z, \boldsymbol{y}_i)$ are pre-calculated and stored in matrix U, the decision z^{SP} can be retrieved efficiently without solving the SAA problem. Moreover, z^{SP} exhibits a different decision-making behaviour compared to z^{SAA}. This difference is illustrated by a PTB example in an ERS problem of a single activity, as shown in Fig. 4. In this example, the task is to allocate an activity to one of the time slots $t \in \{0, 1, 2, 3, 4\}$. The line graph displays the objective values for all feasible decisions across three historical scenarios $Y_{\text{leaf}} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3\}$ and highlights the PTB decisions determined by z^{SP} and z^{SAA}. $\boldsymbol{y}_1$ and $\boldsymbol{y}_3$ share two optimal decisions at time slots $t = 0$ and $t = 1$, while $\boldsymbol{y}_2$ reaches its optimal value at $t = 4$. The decision z^{SAA} at $t = 2$ represents an averaged solution that is robust but not individually optimal, as it fails to reach optimality for any specific scenario. In contrast, z^{SP} at $t = 1$ prioritises achieving optimality for $\boldsymbol{y}_1$ and $\boldsymbol{y}_3$, which results in sacrificing performance of $\boldsymbol{y}_2$.

While averaged results can be more robust for repeated decision-making, the proposed z^{SP} is more focused on finding the optimal solution for the current planning horizon. If z^{SP} is used to compute the loss function during splitting on this Y_{leaf}, it can easily identify the $\boldsymbol{y}_2$ as a 'disruptive instance' that should be excluded from the current split. These make z^{SP} particularly efficient and effective in a rolling-horizon planning setting, where a single optimal solution is required for each planning horizon. Moreover, this example illustrates how z^{SP} prioritises identifying and applying optimal solutions for the historical scenarios within Y_{leaf}. For example, z^{SP} allocates an activity at $t = 1$, which remains optimal for $\boldsymbol{y}_1$ and $\boldsymbol{y}_3$ and has less regrets in $\boldsymbol{y}_2$ than $t = 2$. According to Property 3.2, the ERS problem allows multiple optimal solutions for a single $\boldsymbol{y}_i$, enabling z^{SP} to efficiently identify and transfer a high-quality solution from one historical scenario to another during model training.

Consequently, we replace the z^{SAA} in Eq. (16) by z^{SP}, as shown in Eq. (18). We refer to this loss function as l_{SP}.

$$l_{\text{SP}}(j, s) = \frac{1}{N} \left[\sum_{i \in R_1(j,s)} (\mu(z^{SP}(Y_l), \boldsymbol{y}_i) - u_{ii}) + \sum_{i \in R_2(j,s)} (\mu(z^{SP}(Y_r), \boldsymbol{y}_i) - u_{ii}) \right] \tag{18}$$

The training objective l_{SP} significantly decreases computation time associated with iterative optimisation for z^{SAA}. Fig. 5(a) illustrates how z^{SP} and Eq. (18) are used for decision tree splitting. The candidate splits, generated by the tree splitting function (we adopt the method proposed by Elmachtoub et al. (2020)), divide the adapted scheduling solutions Z_{split} and corresponding objective matrix U_{split} into left and right subsets. We then use l_{SP} to evaluate these splits and select the candidate with the minimum optimisation loss as the new split or leaf node. The primary goal of this process is to identify candidate solutions with a high potential to achieve full-information optimality—that is, solutions exhibit zero regret under the full information condition. As shown in Fig. 5(b), a larger N increases the likelihood of including samples that optimally solve the ERS problem. This broader sample overlap enhances the training process by enabling the decision tree to capture more representative historical scenarios, allowing it to prescribe solutions that more closely approximate full-information optimality.

We can determine the worst-case regret when applying z^{SP} to ERS decision-making, in order to know the worst performance of z^{SP}. To establish this regret, we first consider an energy peak-tariff-free problem as an upper bound. The objective function is simplified to only minimise the energy cost term while maintaining the same constraints as in Eqs. (2)–(13). This simplification allows us to obtain an ERS solution, denoted as z^{ub}, which serves as an upper bound of the original problem. Using this upper bound, we can establish the worst-case regret for z^{SP}, as shown in Property 4.1.

Property 4.1. *The worst-case regret of $z^{SP}(\boldsymbol{y})$, $\forall \boldsymbol{y} \in Y$, is given by*

$\Delta C + \gamma\big(G(z^{SP}, \boldsymbol{y}) - \max(\boldsymbol{y})\big)$, where $\Delta C = C(z^{SP}) - C(z^{ub})$.

Proof. To prove Property 4.1, we first establish the following fundamental property of the ERS problem:

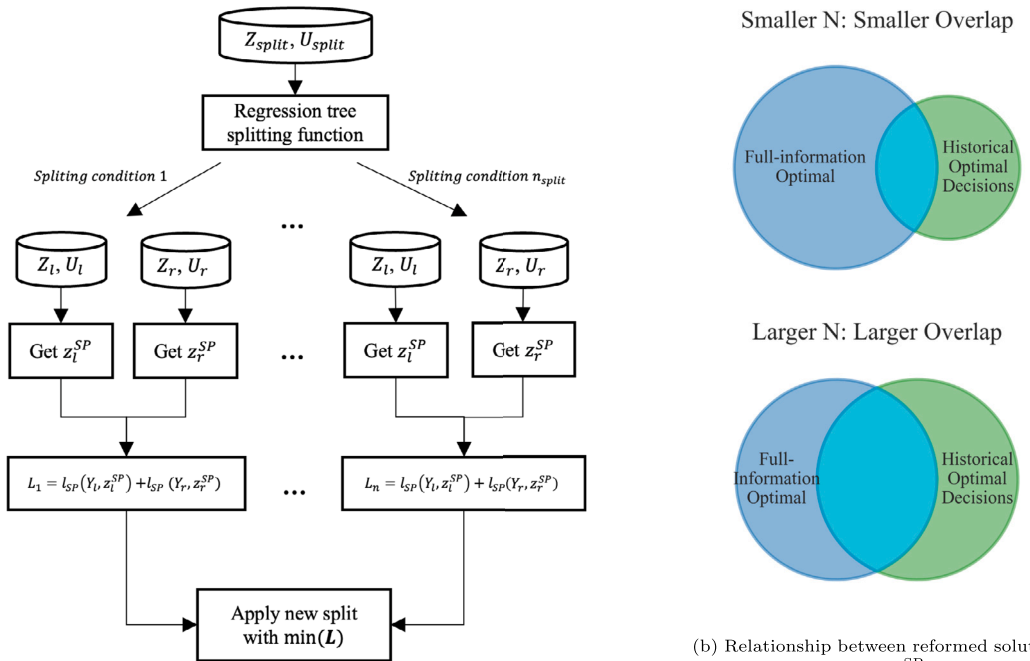
Property 4.2. *The minimum energy demand charge for any ERS solution under an uncertainty array $\boldsymbol{y}$ is $\gamma \cdot max(\boldsymbol{y})$.*

This can be achieved by constructing a solution where no activities are scheduled during the time slot with the highest base energy consumption, $max(\boldsymbol{y})$.

If the solution z^{ub} schedules no activities at the time slot of $G(z^{SP}, \boldsymbol{y})$, the energy demand charge for this scenario will reach its theoretical lower bound, which is $\gamma max(\boldsymbol{y})$; and, given the definition of z^{ub}, its energy cost part $C(z^{ub})$ is optimal. Hence, z^{ub} can be regarded as the full-information optimal solution. Then, for z^{SP}, the worst case value gap in scenario $\boldsymbol{y}$ is given by: $\mu(z^{SP}, \boldsymbol{y}) - \mu(z^{ub}, \boldsymbol{y}) = C(z^{SP}) - C(z^{ub}) + \gamma(G(z^{SP}, \boldsymbol{y}) - \max(\boldsymbol{y}))$, which satisfies the property. □

This property tells when z^{SP} can have the worst performance in the ERS problem. Consequently, we propose Algorithm 1 to train the IPTB model using l_{SP}.

We restructure the greedy learning algorithm for the IPTB model to directly incorporate prescriptive tree-splitting. The algorithm begins by initialising the decision tree and inputting the reformed observation data X, uncertainty data Y, reformed solution data Z and pre-calculated objective value matrix U. During the training of each layer l, the algorithm iterates over all possible splits for each input feature in X. For each candidate split s, decisions $\hat{z}_l$ and $\hat{z}_r$ are computed for the left and right splits by minimising the averaged objective values $u_{ii'} \in U$. The candidate split that minimises the loss function l_{SP} is selected as the optimal split and added to the decision tree model. This iterative process ensures that data samples with similar patterns are grouped together within each split, enhancing the prescriptive power of the model. Moreover, by using pre-calculated objective values stored in U, the algorithm avoids solving the ERS problem for each split, significantly improving training efficiency.



(a) Splitting process by the optimisation loss and z^{SP}.

(b) Relationship between reformed solution data Z (the data of z^{SP}) and the full-information optimal solutions

Fig. 5. The process of the IPTB model optimising decision tree structures by the optimisation loss.

Algorithm 1 The IPTB training algorithm

Input: X, Y, Z, U
  Initialise the decision tree, layer count $l = 1$
  while Maximum depth not reached **do**
    for each split in layer $l-1$ **do**
      for each input feature j in J **do**
        Obtain candidate split set E by splitting function $\hat{\boldsymbol{z}} = \{\}$
        for s in E **do**
          $\hat{z}_l = \arg \min_{i \in R_1(j,s)} \sum_{i' \in R_1(j,s)} u_{ii'}$
          $\hat{z}_r = \arg \min_{i \in R_2(j,s)} \sum_{i' \in R_2(j,s)} u_{ii'}$
          $\hat{\boldsymbol{z}}[j,s] = (\hat{z}_l, \hat{z}_r)$
        end for
      end for
      $(j,s) = \arg \min_{s \in E, j \in J} l_{\mathrm{SP}}(j,s)$
      Update layer l by splitting pair (j,s), $l = l+1$
    end for
  end while
Output the decision tree model

The PTB model's performance can be improved by an ensemble of trees, as proposed by Bertsimas and Kallus (2019). RF models are known to exhibit lower variance compared with individual decision trees, thereby improving model stability and robustness. To construct a prescriptive RF model, multiple PTB models are trained on bootstrapped samples of the training dataset. Algorithm 2 shows the procedure of obtaining decisions by this RF model. Given an observation array $\boldsymbol{x}$, the algorithm iterates over all trees in the ensemble to calculate weight values for each historical scenario i based on the leaf-splitting function $R(\cdot)$. For each tree, $R(\cdot)$ is used to determine the corresponding leaf of $\boldsymbol{x}$, and weights w_i are updated for samples belonging to this leaf. Finally, the solution z^{SP} is computed as the decision that minimises the weighted sum of objective values across all historical scenarios.

Algorithm 2 The procedure of IPTB's decision-making.

Input: learned IPTB-RF model, $\boldsymbol{x}$
  for $tree \in trees$ **do**
    Obtain leaf-splitting function $R()$ from $tree$
    for $i = 1 \ldots N$ **do**
      $w_i += \frac{\mathcal{I}(R(x)=R(x_i))}{|\{j : R(x_j)=R(x)\}|}$
    end for
  end for
  $z^{\mathrm{SP}} = \arg\min_{z \in Z} \sum_{i=1}^{N} w_i(\boldsymbol{x}) \mu(z, \boldsymbol{y}_i)$
Output z^{SP}

After training the IPTB model, we perform tree-structure pruning to remove unnecessary or noisy branches introduced during the greedy tree-building phase. According to Breiman et al. (2017), pruning can be performed using two methods: cost complexity pruning and reduced error pruning. We apply a modified version of the reduced error pruning procedure. Specifically, we calculate the regret of each subtree using z^{SP} as the decision. If replacing a subtree with a single leaf node does not increase the regret, the subtree is pruned and replaced by the leaf node.

5. Experiments and results

This section discusses the experiments conducted and their results. We first carried out a comparison experiment among several ML-integrated DDO methods using randomly generated ERS instances. Various parameter settings of the ERS problem were compared to demonstrate the competitive performance of our approach. Then, we performed a case study on a practical ERS problem, utilising real-world ERS instances and energy price data. The uncertain and contextual datasets were derived from a technical challenge (Bergmeir, 2021). Our implementation was done using Python 3.9, and the ERS problem was solved using Gurobi 10.0.1. All experiments were conducted on a 64-core server running at 2.6 GHz with 64 GB of RAM.

5.1. Experimental data

The data originated from an IEEE-CIS technical challenge (Bergmeir, 2021). Uncertainty data, Y, were collected from six buildings located at Monash University, Clayton Campus, with measurements taken at 15-minute intervals (each $t \in T$ represents a 15-minute segment) from January 2016 to November 2020. The power consumption of all buildings and the power generated by their solar panels were recorded every 15 min, which resulted in a training dataset with 705,818 rows. For the contextual data, X, this study used two open-access time–series weather datasets from oikolab (2022) and the Australian Bureau of Meteorology, along with calendar data for the corresponding period. The 17 weather-related features include variables such as temperature, dew point temperature, wind speed, mean sea-level pressure, relative humidity, surface solar radiation and surface thermal radiation. We applied the data reforming approach proposed in Section 4.1 to transfer the dataset into a prescriptive dataset and pre-process it. Regarding the energy price data, we applied different energy price data for the experiments. In the randomly generated experiments, we used a time-of-use energy price distribution from an energy-aware scheduling problem (Luo et al., 2013), as expressed in Eq. (19) below:

$$\text{price} = \begin{cases} 50 & \text{if hour} < 3 \text{ or hour} > 21 \\ 100 & \text{if } 3 \leq \text{hour} \leq 17 \\ 80 & \text{if } 17 < \text{hour} \leq 21 \end{cases} \tag{19}$$

In the case study, we applied the energy data collected from the Australian Energy Market Operator, which comprises half-hourly electricity price and demand data at the state level.

5.2. Comparison of different ML-integrated data-driven optimisation methods

In the first experiment, the proposed IPTB model was applied for decision-making in the ERS problem, and was compared with several modern ML-integrated DDO methods, namely PO, SAA, and PTB. Firstly, SAA was applied as a stochastic optimisation solution that is independent of contextual data X. We then compared two different methods that include foundational elements of the proposed IPTB model:

- PO-DT/RF: PO solutions with tree-based models as its predictors.
- PTB-DT/RF: Prescriptive solutions prescribed by PTB models (Bertsimas & Kallus, 2019).

We set the hyperparameters of all tree-based models equally to ensure a fair comparison. Specifically, the number of trees in each forest was set to 200, with no limitation on the maximum tree depth, and each leaf node must contain at least 5 data instances. The forests were constructed using a bootstrap method combined with a subsampling technique, where each tree was trained on 80% of the data samples. Unlike existing research that often limits the number of possible splits for efficiency, such as using a fixed number of candidate splits (Elmachtoub & Grigas, 2021) or keeping a minimum percentage of data in the split, we adapted an exhaustive search method. This approach ensures that all possible splits are considered at each node, preventing the best possible partition from being missed during the tree-building process.

5.2.1. Comparison of different problem sizes

We first generated multiple ERS instances by varying n_a values, which range from $n_a = 20$ to $n_a = 80$ in increments of 20. Here, $n_a = 20$ represents a relatively loose timetable, while $n_a = 80$ is considered a challenging benchmark instance, according to project scheduling benchmarks (Kolisch & Sprecher, 1997). To ensure that the generated ERS instances possess the desired attributes, we designed instances that are close to parallel and that feature a combination of long activity chains and independent activities (Bergmeir et al., 2022). This approach helps prevent superficial hardness characteristics that might emerge in randomly generated instances (Vanhoucke et al., 2008).

The optimisation performance was tested using 50 generated instances. We evaluated all scheduling solutions for the testing instances based on the objective value loss $l_{obj} = \mu(\mathbf{y}, \hat{z}_i) - \mu(\mathbf{y}, z_i^*)$ and regret percentage $regret = \frac{\mu(\mathbf{y},\hat{z}_i) - \mu(\mathbf{y},z_i^*)}{\mu(\mathbf{y},z_i^*)} \times 100$. Here, $\hat{z}_i$ is the applied solution for the ith instance, and z_i^* represents the optimal solution under full information for the ith instance. We also use a box plot to visually summarise the outcome distributions. Result points outside 1.5 times the interquartile range are considered outliers and are denoted by circles.

Table 3 and Fig. 6 indicate that the proposed IPTB model significantly outperforms all benchmark models across all n_a settings. When n_a is small (e.g., $n_a = 20$ and $n_a = 40$), most of the decisions made by the IPTB-RF model reach full-information optimality, which result in 0% regrets. As n_a increases to 60 and 80, although the IPTB-RF could not prescribe optimal scheduling solutions, it still performs the best compared to other algorithms. Among the RF-based methods, PTB-RF exhibits the second-best performance based on averaged and median regret values, while SAA-RF and PO-RF rank third and fourth, respectively. PO-RF has a lower mean absolute percentage loss of 0.075 than PO-DT's 0.096, based on our prediction results. However, it incurs greater scheduling losses than PO-DT in all scenarios. This counter-intuitive outcome underscores that prediction errors do not necessarily correlate with downstream optimisation performance in the ERS problem. The PTB-DT method has the poorest performance among all algorithms, suggesting that models trained exclusively on prediction loss are inadequate for addressing this problem. Our IPTB approach not only surpasses other decision tree models but also outperforms the SAA method, demonstrating its effectiveness in solving the stochastic ERS problem.

5.2.2. Comparison of different γ

To uncover different decision-making behaviours of the compared models, we varied key endogenous factors in this experiment. Specifically, we analysed the IPTB model's performance under different γ settings, where γ controls the relative importance of energy demand charge in the objective function. Apart from the tested $\gamma = 0.1$ setting, we evaluated three other settings: small $\gamma = 0.05$, normal $\gamma = 5$ and large $\gamma = 50$. When $\gamma = 5$, the importance of energy demand charge and energy cost is identical; when $\gamma = 50$, the model first tries to minimise the energy demand charge term and then the energy cost. To balance the model complexity, we used the 50 ERS instances that were generated with $n_a = 60$ from the previous experiment (i.e, a medium-level setting for n_a).

According to Fig. 7 and Table 4, the energy demand charge parameter γ significantly affects both problem difficulty and model performance. The ranking of models is consistent with the results of previous experiments: the IPTB-RF model consistently outperforms other models, with PTB-RF ranking second. Notably, when γ is set to smaller values ($\gamma = 0.05$ and $\gamma = 0.1$), the PO solutions are still better than the SAA solutions. However, as γ increases to 5 and 50, the performance of these PO solutions declines greatly, resulting in the model-free solution SAA to outperform them. This observation indicates that, despite having relatively precise predictions, PO fails to yield good optimisation results for this ERS problem. According to Eq. (2), the energy cost is a linear function where γ controls the significance of the energy demand charge. A higher γ means that even a small prediction error for large net energy consumption values can result in a significant optimisation loss, while predictions for other time slots become less critical. PO treats all time slots equally during prediction, thereby ignoring the importance of peak-time energies. This can result in greater losses when γ is high.

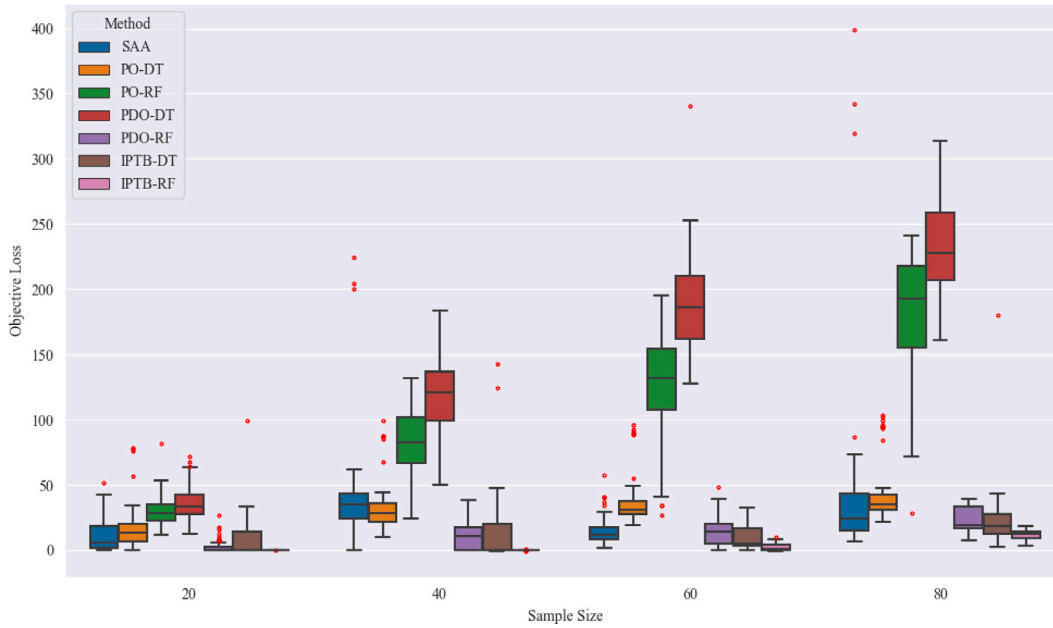
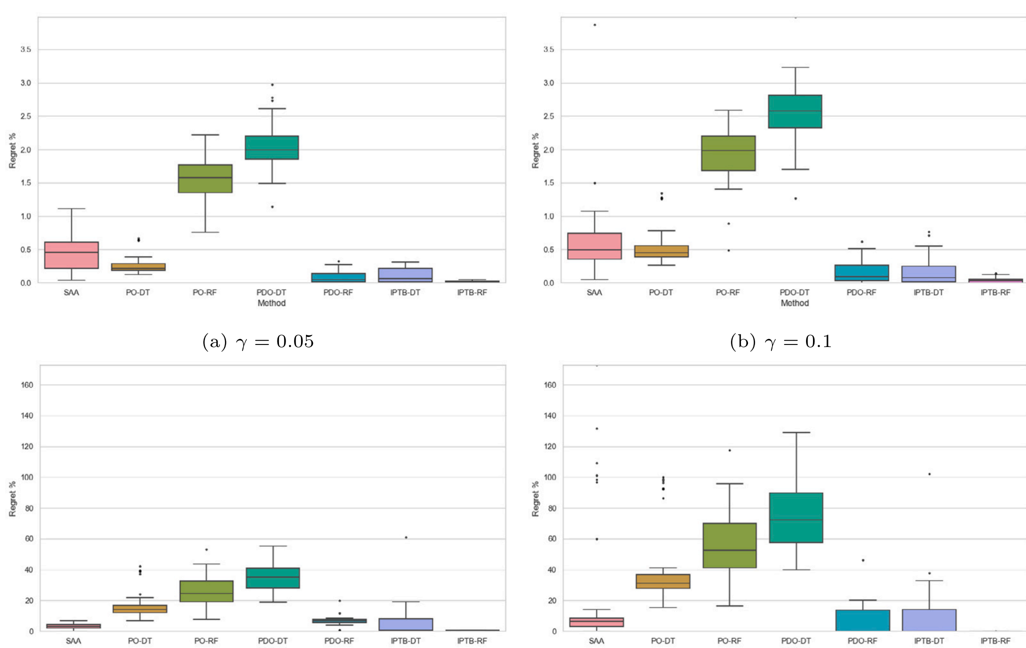


Fig. 6. Objective loss comparison of different ML-DDO methods by different n_as.

Table 3
Averaged regret percentages of different ML-DDO methods by varying n_as.

	Model	$n_a = 20$	$n_a = 40$	$n_a = 60$	$n_a = 80$
Model-free	SAA	11.23%	43.45 %	14.58 %	48.19 %
Tree	PO-DT	16.47%	33.62%	38.36%	44.66%
	PTB-DT	36.66%	117.77%	190.43%	230.68%
	IPTB-DT	**8.58%**	**14.75%**	**9.59%**	**22.83%**
Forest	PO-RF	30.02%	83.36%	126.51%	184.32%
	PTB-RF	2.69%	10.81%	13.51%	23.06%
	IPTB-RF	**0.00%**	**0.02%**	**2.09%**	**11.55%**



(a) $\gamma = 0.05$

(b) $\gamma = 0.1$

(c) $\gamma = 5$

(d) $\gamma = 50$

Fig. 7. Box plots of different ML-DDO methods under various γ settings.

Table 4
Averaged regret percentages of different ML-DDO methods under various γ settings.

	Model	$\gamma = 0.05$	$\gamma = 0.1$	$\gamma = 5$	$\gamma = 50$
Model-free	SAA	0.45%	0.61%	3.16%	19.91%
Tree	PO-DT	0.27%	0.55%	16.86%	41.72%
	PTB-DT	2.02%	2.57%	35.33%	74.96%
	IPTB-DT	**0.11%**	**0.16%**	**5.16%**	**7.79%**
Forest	PO-RF	1.56%	1.94%	26.11%	56.45%
	PTB-RF	0.09%	0.16%	6.31%	6.38%
	IPTB-RF	**0.01%**	**0.04%**	**0.40%**	**0.16%**

Table 5
Runtime and regret comparison of tree models trained by optimisation loss.

	Model	No. of leaves	Runtimes (s)	Regret %
Small problem	PTB-PL	180.4	**0.1**	2.57%
	DT-SPO	123.1	3196.3	0.53%
	PTB-SPO	112.2	5125.1	0.26%
	IPTB-DT	115.0	76.3	**0.26%**
Large problem	PTB-PL	179.2	**0.2**	35.33%
	DT-SPO	67.3	OT	10.24%
	PTB-SPO	25.2	OT	**4.78%**
	IPTB-DT	137.8	94.1	5.27%

In contrast, the prescriptive methods capture sufficient energy-peak patterns for optimisation, demonstrating better performance as γ increases. The standout performance of our IPTB approach is particularly evident when $\gamma = 50$. While other methods suffer greater performance losses, the IPTB-RF model shows a performance gain compared to $\gamma = 5$. Upon examining the results, we find that the IPTB-RF model achieves 18 full-information optimal solutions among all 50 instances These optima are achieved by leveraging alternative optimal solutions within the reformed solution data Z, demonstrating that the optimal search capability highlighted in Fig. 4 remains effective even under high γ settings. In contrast, when using z^{PTB} in the PDO-RF framework, fewer scenarios reach the optimal state, with only 3 out of 50 instances achieving it. Although the weighted SAA approach guarantees robust average performance, it does not identify as many optimal solutions. According to Property 4.1, the minimum energy demand charge is $max(\boldsymbol{y})$. When $\gamma = 50$, optimising activity energies during the time slot of $max(\boldsymbol{y})$ becomes more critical than optimising the energies of other activities. This highlights that IPTB excels in finding scheduling solutions that minimise high-energy peaks. Such capability is particularly valuable in practical applications where compact ERS scenarios demand precise energy management of peak loads.

We also note that the PTB-RF method has exhibited unstable performance with increasing γ, though it performs stably when n_a increases. This underscores the importance of using optimisation loss to identify suitable scheduling samples, particularly when uncertainty significantly impacts the constraint space. From a practical perspective, increasing γ addresses the importance of energy peak demand and ensures more balanced energy use by leveraging renewable energies. This setting becomes increasingly practical in real-world ERS applications due to the rising demand for sustainability. When addressing a real-world ERS problem that prioritises balanced energy demand, the proposed methodologies will effectively balance the peak-time energy consumption with overall energy cost, thereby producing more stable and reliable outcomes.

5.3. Comparison of training loss functions

To further explore the benefits of applying optimisation loss to ERS, we compare different ILO approaches with three training loss functions under both PO and prescriptive paradigms, including:

- Decision tree trained by Eq. (15) (DT-SPO);
- PTB trained by prediction loss (PTB-PL);
- PTB trained by modified loss Eq. (16) (PTB-SPO);
- PTB trained by proposed loss function Eq. (18) (IPTB-DT).

Both model efficiency and effectiveness were evaluated by comparing all approaches' number of leaves, runtimes and regret metrics. Since Eq. (15) and Eq. (16) require calculating the optimisation problem during training, the ERS problem size must be restricted. Otherwise, the first two approaches cannot be trained within an acceptable time period. We set the maximum training time to 7,200 s. Once the training limit is reached, the training process ends, and the trained model is used to directly predict/prescribe the tested ERS solutions. Based on previous experiments, we used two different ERS problem settings to create 20 ERS instances: (1) small problem $n_a = 40$ and $\gamma = 0.1$; and (2) large problem $n_a = 60$ and $\gamma = 5$.

As Table 5 illustrates, the IPTB method demonstrates a significant advantage over models trained using the SPO loss function, delivering better performance in both runtimes and regret percentages. However, the runtime metrics vary considerably. For smaller problem sizes, IPTB achieves a regret of 0.26% in 76.3 s, markedly outperforming PTB-SPO, which has a similar regret of 0.26% but requires 5125.1 s. For larger problem sizes, the performance gap widens, with IPTB completing in 94.1 s with a regret of 5.10%, whereas SPO loss-trained models exceed the 2-hour limit without completing the task. The PTB-PL method, while showing the fastest runtimes (0.1 s for small problems and 0.2 s for large problems), suffers from higher regret percentages, particularly in larger problems (35.33%). This highlights the trade-off between runtime efficiency and model performance, where IPTB strikes a balance by achieving both lower regret and reasonable training times. Additionally, the IPTB training time remains relatively insensitive to problem sizes, showing only a 20-second increase for larger problems. This efficiency is due to the IPTB approach leveraging a pre-calculated matrix U to accelerate training.

The ability of IPTB to utilise the pre-calculated matrix U efficiently underscores its robustness and scalability in solving complex energy scheduling problems. This approach proves effective in approximating sub-optimal weighted SAA solutions through the ML model, indicating that IPTB's final solution quality remains high across varying problem sizes and conditions.

5.4. Case study: ERS problem instances with real-world data

We finally conducted a case study focusing on real-world ERS datasets from IEEE-CIS's PO technical challenge (Bergmeir, 2021). The goal is to schedule a set of activities across six buildings in a way that

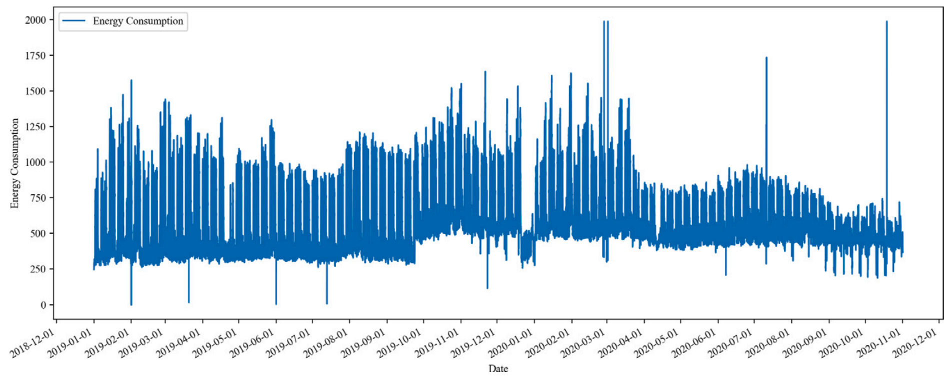


Fig. 8. Plot of the training and testing data.

minimises total energy costs and energy demand charges. We applied the proposed IPTB model to directly address the optimisation task and compared various MLDDO methods to evaluate their performance. The challenge offers two instance sizes, with five instances for each size. Small instances $SI0 - SI4$ comprise 50 activities, which represent a solvable n_a setting. Conversely, large instances $LI0 - LI4$ contain 200 activities, which are unlikely to be solved optimally using general optimisation solvers. We used the data from January 2019 to October 2020 for the training set, and data from November 2020 for the testing set. Fig. 8 illustrates the distribution of overall energy consumption data, indexed by 15-minute slots. This overall consumption is the total building energy consumption minus the total solar panel's energy generation, having an identical definition as y_t described in Section 3. The average energy consumption is 608.07 units, with a standard deviation of 248.84, while the testing average is 557.58, with a standard deviation of 152.05.

For comparison, we first used two practical methodologies that have demonstrated strong performance in the challenge. The first methodology, SAA-LGBM, includes an SAA-based approach that applies six light gradient boost machine models as prediction models. This approach involves using both hourly and daily data to train different light gradient boost machine models. We employed the six predictions for SAA computations of $SI0 - SI4$ and $LI0 - LI4$ to generate final decisions. Another methodology, PO-RF, includes a general PO approach with a well-tuned RF model that is trained with additional data to improve accuracy. This prediction was directly applied to $SI0-SI4$ and $LI0-LI4$ as input parameters to conduct PO. Lastly, we compared the prediction loss trained PTB model to see the gain of applying true optimisation loss in this problem.

As indicated by Limmer and Einecke (2022), the small instances in this case study can be solved optimally within an acceptable time period. However, due to the significantly larger problem size, obtaining optimal solutions for the monthly large-scale instances is impossible. Consequently, we used near-optimal solutions to train the monthly IPTB model by setting the gap percentage of the Gurobi solver to 5% instead of pursuing global optimal solutions.[1] This approach also simulates the application of heuristics with a worst-case gap to the proposed IPTB model, helping us evaluate our model's robustness with practical optimisation techniques. The original IEEE-CIS technical challenge encompasses a more complex problem that considers a quadratic energy demand charge term. However, in practical scenarios, the energy demand charge is usually linear to balance the time-of-use energy charge and peak-time energy demand (Jiang et al., 2021). Considering that the quadratic term increases the difficulty of obtaining optimal solutions, even with complete information on the uncertainties. In this paper, we focused on a prescriptive methodology; we thus considered a simplified model without losing feasibility for the original problem. Additionally, to further investigate the fitness of each method for both the observation data X and ERS instances, we compared the averaged SPO loss and prescriptiveness coefficient proposed by Bertsimas and Kallus (2019). This coefficient is a unitless quantity bounded above by 1. Specifically, a value of 0 implies that the DDO decision z is equivalent to the model-free SAA decision, while a value of 1 indicates that the decision matches the perfect-foresight optimal decision z^*. Negative values suggest that the DDO decision is worse than the SAA decision. Therefore, a prescriptiveness coefficient below 0 indicates that the DDO method, along with the observation data X, fails to fit the ERS instance well, as it underperforms compared to the model-free SAA decision.

Table 6 presents the results of monthly schedules, in which it compares SPO loss values from October and November. Values in bold font highlight the method that exhibits the best performance among all comparisons. In the October results, the proposed IPTB model outperforms all benchmark methods, except for $LI0$ and $LI4$. It surpasses the PTB model by 22.13%, 39.53%, 71.67%, 10.06%, and 63.28% in small instances, and shows improvement by 9.57%, 64.87%, 16.52%, 23.56% and 1.97% in large instances. Both the PO-RF and SAA-LGBM methods underperform in comparison to the prescriptive models. Regarding the November schedules, the IPTB model surpasses other models in most cases, even when near-optimal solutions are employed during training. In small instances, the PTB model ranks second; however, its performance declines in larger instances, falling behind the PO-RF model. The IPTB model specifically demonstrates 3.11%, 6.94%, 56.82%, 40.88% and 44.69% improvements over the PTB model from large instances $LI0$ to $LI4$, respectively. According to these results, although PTB and IPTB are both prescriptive methods, the PTB model exhibits a more significant performance drop, mainly because of its reliance on the prediction loss function for training. In contrast, the IPTB model maintains stability and prescriptiveness across all November instances and most October instances, thereby solidifying its effectiveness and robustness in this case study.

We further analysed the prescriptiveness coefficients, as depicted in Table 7. Values in bold highlight the best-performing method. In October, the proposed IPTB method displays a stable prescriptiveness performance on the small instances: the averaged prescriptiveness is higher than 0.5, which indicates that it reaches halfway between the method-free SAA decision and full-knowledge optimal. The PTB and PO methods display instability in SI_2 and SI_4, with prescriptiveness values falling below 0. Moreover, the SAA method underperforms compared to all other methods. In contrast, the scenario changes on the larger instances in October. PO emerges as the top performer for LI_0 and LI_4 in October. Additionally, more instances in October revealed prescriptiveness coefficients that are lower than 0. As mentioned before, a negative coefficient of prescriptiveness value indicates inadequate fitting of observation data X to the method and ERS instance. This represents a data-scarce scenario in which the observation

[1] The actual gaps can be smaller, as this gap is a worst-case gap.

Table 6
SPO losses of different methods in the case study experiment.

	October				November			
	SAA-LGBM	PO-RF	PTB	IPTB	SAA-LGBM	PO-RF	PTB	IPTB
SI_0	6.46%	4.29%	3.50%	**2.73%**	19.94%	12.29%	11.93%	**9.21%**
SI_1	22.73%	11.30%	10.77%	**6.51%**	9.21%	3.10%	3.06%	**3.00%**
SI_2	20.58%	13.32%	16.96%	**4.81%**	20.17%	12.65%	12.00%	**6.79%**
SI_3	16.81%	10.82%	10.65%	**9.58%**	**9.81%**	10.52%	10.53%	**9.81%**
SI_4	18.34%	14.73%	16.71%	**6.14%**	1.05%	1.24%	1.53%	**0.91%**
LI_0	10.84%	**8.56%**	11.06%	10.01%	38.32%	25.54%	20.88%	**20.23%**
LI_1	15.69%	13.18%	13.18%	**4.63%**	24.08%	11.21%	11.66%	**10.86%**
LI_2	13.93%	10.34%	9.24%	**7.71%**	17.16%	13.02%	13.08%	**5.65%**
LI_3	12.15%	11.88%	9.19%	**7.02%**	12.99%	4.98%	4.71%	**2.78%**
LI_4	8.57%	**4.58%**	7.20%	7.06%	13.24%	13.70%	13.28%	**7.35%**

Table 7
Comparison of coefficient of prescriptiveness.

	October				November			
	SAA-LGBM	PO-RF	PTB	IPTB	SAA-LGBM	PO-RF	PTB	IPTB
SI_0	0.09	0.38	0.38	**0.61**	−0.13	0.16	0.25	**0.37**
SI_1	−0.42	0.29	0.33	**0.59**	−0.61	0.22	0.25	**0.27**
SI_2	−2.19	−1.29	−0.61	**0.45**	−0.63	−0.24	−0.24	**0.46**
SI_3	0.12	0.28	0.29	**0.36**	0.30	0.08	0.10	**0.31**
SI_4	−0.71	−0.37	−0.56	**0.43**	0.16	0.16	0.10	**0.20**
LI_0	−0.59	**0.22**	−0.60	−0.59	−0.35	−0.08	0.08	**0.20**
LI_1	−0.70	−0.84	−0.75	**−0.31**	−0.65	0.32	0.28	**0.48**
LI_2	0.14	0.14	0.45	**0.48**	−0.10	0.05	0.10	**0.39**
LI_3	−0.12	0.21	0.35	**0.65**	0.14	0.35	0.35	**0.58**
LI_4	−0.27	**−0.08**	−0.31	−0.30	−0.54	−0.64	−0.54	**0.46**

data X does not adequately fit the large ERS instances in October 2020. Without sufficient data, the accuracy of the ML models using near-optimal scheduling solutions could be significantly reduced. Despite this, our proposed IPTB model still outperforms other methods for the October instances LI_1, LI_2, and LI_3. When leveraging DDO methods with inadequate datasets, our methods can still produce a robust schedule, resulting in more stable performance. Regarding the November instances, we note that the observation dataset X is better suited for solving the ERS problem, resulting in fewer instances of prescriptiveness coefficients that are lower than 0. In this scenario, the IPTB model trained with near-optimal solutions yields an even more stable result. Not only does it improve performance more than two-fold when compared with the second-best methods, it also does not present any instances involving prescriptiveness coefficients that are lower than 0—even in cases such as SI_2 and LI_4, in which all other methods exhibit prescriptiveness coefficients that are lower than 0. This implies that even using near-optimal scheduling solutions, our model can still achieve sufficient prescriptiveness compared to optimal solutions ($SI_0 - SI_4$). This flexibility with weaker oracles allows our methodology to be applied across a wide range of applications, without the need for specifying optimisation solutions.

We conducted a deeper analysis of the uncertainty pattern and demonstrated how the proposed IPTB model addresses the data scarcity issue in the November instances. According to Fig. 8, the energy consumption patterns in October and November exhibit distinct and novel behaviours compared with the historical data. Consequently, solving the scheduling problem for October becomes challenging because of the unfamiliar pattern. It is worth noting that this case study was conducted during the COVID-19 pandemic, which introduces additional factors and uncertainties that are not considered in the observation data. In such situations, in which the pattern is new and undiscovered features are involved, a more robust decision-making process is crucial for producing stable outcomes. Therefore, the IPTB model outperforms other methods in most October instances. According to Fig. 8, both November and October share similar results regarding energy consumption. During the IPTB process in November, the October data is reformed using the proposed data construction method to fit the November instances. Given the availability of October data in the November case study, the proposed method can efficiently learn the new pattern and ensure a high level of prescriptiveness. It captures the energy consumption pattern observed in October and uses the pattern for decision-making, which results in a higher prescriptiveness coefficient in November. In summary, the proposed IPTB method ensures a robust schedule, even with limited datasets; therefore, it proves its efficacy and adaptability in terms of solving the ERS problem.

6. Conclusion and future work

In this study, we investigated the ERS problem with uncertainties relating to the energy consumption and renewable energy generation of buildings. We proposed an IPTB model to solve this problem efficiently. Leveraging the ERS problem's property, we ensured the tractability of the proposed IPTB model, and we presented a new loss function l_{SP} to integrate the optimisation problem with the ML model. Additionally, we developed techniques to optimise the IPTB's parameters and prune its structure to improve learning efficiency and prevent overfitting.

To evaluate the IPTB model's performance, we conducted experiments using randomly generated datasets. By varying the problem size, we demonstrated that the proposed IPTB model could solve a large-scale ERS problem accurately. We also examined the impact of the energy demand charge parameter γ and proposed strategies for ERS decision-making when energy demand charge weights are high. Further, we carried out a case study on a real-world ERS problem. The results revealed that the IPTB model can handle complex constraints and prescribe robust schedules, even when the observation data only partially fits the ERS problem.

Despite our promising results, certain aspects warrant further investigation. In this study, we set the gap percentage of the optimisation solver to simulate the IPTB model's performance with heuristics. The real performance of the proposed IPTB model with heuristics has not been tested. Additionally, this study utilised a grid search-based method for hyper-parameter tuning, prioritising time efficiency. A more advanced parameter tuning algorithm for the proposed algorithm could significantly improve prescriptiveness. Therefore, future research focusing on developing specialised hyper-parameter tuning algorithms for prescriptive models is highly promising.

CRediT authorship contribution statement

Siping Chen: Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Raymond Chiong:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Investigation, Conceptualization. **Debiao Li:** Writing – review & editing, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization.

Acknowledgements

Siping Chen is sponsored by funding from the China Scholarship Council, China (Grant No. 202106650014). Debiao Li's work is supported by funding from the National Natural Science Foundation of China (Grant No. 72171054) and the Fujian Provincial Natural Science Foundation, China (Grant No. 2023J06015).

References

Abedi, M., Chiong, R., Noman, N., Liao, X., & Li, D. (2024). A metaheuristic framework for energy-intensive industries with batch processing machines. *IEEE Transactions on Engineering Management*, *71*, 4502–4516.

Akbarzadeh, B., & Maenhout, B. (2021). A decomposition-based heuristic procedure for the Medical Student Scheduling problem. *European Journal of Operational Research*, *288*(1), 63–79.

Bagger, N.-C. F., Sørensen, M., & Stidsen, T. R. (2019). Dantzig–Wolfe decomposition of the daily course pattern formulation for curriculum-based course timetabling. *European Journal of Operational Research*, *272*(2), 430–446.

Bartusch, M., Mohring, R. H., & Radermacher, F. J. (1988). Scheduling project networks with resource constraints and time windows. *Annals of Operations Research*, *16*(1–4), 201–240.

Bergmeir, C. (2021). *IEEE-CIS Technical Challenge on Predict+Optimize for Renewable Energy Scheduling*. IEEE Dataport.

Bergmeir, C., de Nijs, F., Sriramulu, A., Abolghasemi, M., Bean, R., Betts, J., Bui, Q., Dinh, N. T., Einecke, N., Esmaeilbeigi, R., Ferraro, S., Galketiya, P., Genov, E., Glasgow, R., Godahewa, R., Kang, Y., Limmer, S., Magdalena, L., Montero-Manso, P., ..., Peralta, D. et al. (2022). Comparison and evaluation of methods for a Predict+Optimize problem in renewable energy. In: arXiv http://dx.doi.org/10.48550/arXiv.2212.10723.

Bertsimas, D., & Kallus, N. (2019). From predictive to prescriptive analytics. *Management Science*, *66*, 1025–1044.

Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. J. (2017). Classification and Regression Trees. New York: Chapman and Hall/CRC, ISBN: 978-1-315-13947-0.

Cacchiani, V., Caprara, A., Roberti, R., & Toth, P. (2013). A new lower bound for curriculum-based course timetabling. *Computers & Operations Research*, *40*(10), 2466–2477.

Cardoen, B., Demeulemeester, E., & Beliën, J. (2010). Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, *201*(3), 921–932.

Carriere, T., & Kariniotakis, G. (2019). An integrated approach for value-oriented energy forecasting and data-driven decision-making application to renewable energy trading. *IEEE Transactions on Smart Grid*, *10*(6), 6933–6944.

Demirovi, E., Stuckey, P., Guns, T., Bailey, J., Leckie, C., Ramamohanarao, K., & Chan, J. (2020). Dynamic programming for Predict+Optimise. In *Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 02* (pp. 1444–1451).

Department of Climate Change, Energy, the Environment, and Water (2022). *Commercial Buildings Energy Consumption Baseline Study 2022*: *Technical Report*, Australian Government.

Dudek, G., Pełka, P., & Smyl, S. (2021). A hybrid residual dilated LSTM and exponential smoothing model for midterm electric load forecasting. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(7), 2879–2891.

Elmachtoub, A. N., & Grigas, P. (2021). Smart "Predict, then Optimize. *Management Science*, *68*, 9–26.

Elmachtoub, A. N., Liang, J. C. N., & McNellis, R. (2020). Decision trees for decision-making under the predict-then-optimize framework. In *Proceedings of Machine Learning Research*: *vol. 119*, *Proceedings of the 37th International Conference on Machine Learning* (pp. 2858–2867).

Escrivá-Escrivá, G., Segura-Heras, I., & Alcázar-Ortega, M. (2010). Application of an energy management and control system to assess the potential of different control strategies in HVAC systems. *Energy and Buildings*, *42*(11), 2258–2267.

Esmaeilbeigi, R., Mak-Hau, V., Yearwood, J., & Nguyen, V. (2022). The multiphase course timetabling problem. *European Journal of Operational Research*, *300*(3), 1098–1119.

Fadzli Haniff, M., Selamat, H., Yusof, R., Buyamin, S., & Sham Ismail, F. (2013). Review of HVAC scheduling techniques for buildings towards energy-efficient and cost-effective operations. *Renewable and Sustainable Energy Reviews*, *27*, 94–103.

González-Torres, M., Pérez-Lombard, L., Coronel, J. F., Maestre, I. R., & Yan, D. (2022). A review on buildings energy information: Trends, end-uses, fuels and drivers. *Energy Reports*, *8*, 626–637.

Hong, T., Pinson, P., Fan, S., Zareipour, H., Troccoli, A., & Hyndman, R. J. (2016). Probabilistic energy forecasting: Global Energy Forecasting Competition 2014 and beyond. *International Journal of Forecasting*, *32*(3), 896–913.

Hu, T., Guo, Q., Li, Z., Shen, X., & Sun, H. (2019). Distribution-free probability density forecast through deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(2), 612–625.

Jeong, J., Jaggi, P., Butler, A., & Sanner, S. (2022). An exact symbolic reduction of linear smart predict+optimize to mixed integer linear programming. In *Proceedings of Machine Learning Research*: *vol. 162*, *Proceedings of the 39th International Conference on Machine Learning* (pp. 10053–10067).

Jiang, Z., Risbeck, M. J., Ramamurti, V., Murugesan, S., Amores, J., Zhang, C., Lee, Y. M., & Drees, K. H. (2021). Building HVAC control with reinforcement learning for reduction of energy cost and demand charge. *Energy and Buildings*, *239*, Article 110833.

Kallus, N., & Mao, X. (2023). Stochastic optimization forests. *Management Science*, *69*(4), 1975–1994.

Khaniyev, T., Kayış, E., & Güllü, R. (2020). Next-day operating room scheduling with uncertain surgery durations: Exact analysis and heuristics. *European Journal of Operational Research*, *286*(1), 49–62.

Khodayar, M., Liu, G., Wang, J., Kaynak, O., & Khodayar, M. E. (2020). Spatiotemporal behind-the-meter load and PV power forecasting via deep graph dictionary learning. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(10), 4713–4727.

Kolisch, R., & Sprecher, A. (1997). PSPLIB - A project scheduling problem library: OR Software - ORSEP Operations Research Software Exchange Program. *European Journal of Operational Research*, *96*(1), 205–216.

Kotary, J., Fioretto, F., Van Hentenryck, P., & Wilder, B. (2021). End-to-end constrained optimization learning: A survey. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence* (pp. 4475–4482).

Li, G., & Chiang, H.-D. (2016). Toward cost-oriented forecasting of wind power generation. *IEEE Transactions on Smart Grid*, *9*(4), 2508–2517.

Limmer, S., & Einecke, N. (2022). An efficient approach for peak-load-aware scheduling of energy-intensive tasks in the context of a public IEEE challenge. *Energies*, *15*(10), Article 3718.

Liu, H., & Grigas, P. (2021). Risk bounds and calibration for a smart predict-then-optimize method. In *Advances in Neural Information Processing Systems*: *vol. 34*, (pp. 22083–22094).

Lu, Z., Cui, W., & Han, X. (2015). Integrated production and preventive maintenance scheduling for a single machine with failure uncertainty. *Computers & Industrial Engineering*, *80*, 236–244.

Luo, H., Du, B., Huang, G. Q., Chen, H., & Li, X. (2013). Hybrid flow shop scheduling considering machine electricity consumption cost. *International Journal of Production Economics*, *146*(2), 423–439.

Mandi, J., Demirovi, E., Stuckey, P. J., & Guns, T. (2020). Smart Predict-and-Optimize for hard combinatorial optimization problems. In *Proceedings of the AAAI conference on Artificial Intelligence, vol. 34* (pp. 1603–1610).

oikolab (2022). oikolab: Weather and climate data for analysts. https://oikolab.com.

Pogancic, M. V., Paulus, A., Musil, V., Martius, G., & Rolinek, M. (2020). Differentiation of blackbox combinatorial solvers. In *Proceedings of the International Conference on Learning Representations*.

Sadana, U., Chenreddy, A., Delage, E., Forel, A., Frejinger, E., & Vidal, T. (2024). A survey of contextual optimization methods for decision-making under uncertainty. *European Journal of Operational Research*, *320*(2), 271–289.

Santamouris, M., & Vasilakopoulou, K. (2021). Present and future energy consumption of buildings: Challenges and opportunities towards decarbonisation. *E-Prime-Advances in Electrical Engineering, Electronics and Energy*, *1*, Article 100002.

Sethanan, K., Theerakulpisut, S., & Benjapiyaporn, C. (2014). Improving energy efficiency by classroom scheduling: A case study in a Thai University. *Advanced Materials Research*, *931–932*, 1089–1095.

Shao, K., Fan, W., Lan, S., Kong, M., & Yang, S. (2023). A column generation-based heuristic for brachytherapy patient scheduling with multiple treatment sessions considering radioactive source decay and time constraints. *Omega*, *118*, Article 102853.

Song, K., Kim, S., Park, M., & Lee, H.-S. (2017). Energy efficiency-based course timetabling for university buildings. *Energy*, *139*, 394–405.

Stratigakos, A., Camal, S., Michiorri, A., & Kariniotakis, G. (2022). Prescriptive trees for integrated forecasting and optimization applied in trading of renewable energy. *IEEE Transactions on Power Systems*, *37*(6), 4696–4708.

Tian, X., Yan, R., Liu, Y., & Wang, S. (2023). A smart predict-then-optimize method for targeted and cost-effective maritime transportation. *Transportation Research Part B: Methodological*, *172*, 32–52.

Vanhoucke, M., Coelho, J., Debels, D., Maenhout, B., & Tavares, L. V. (2008). An evaluation of the adequacy of project network generators with systematically sampled networks. *European Journal of Operational Research*, *187*(2), 511–524.

Yan, R., Wang, S., & Fagerholt, K. (2020). A semi-"smart predict then optimize" (semi-SPO) method for efficient ship inspection. *Transportation Research Part B: Methodological*, *142*, 100–125.

Yang, Z., & Becerik-Gerber, B. (2014). The coupled effects of personalized occupancy profile based HVAC schedules and room reassignment on building energy use. *Energy and Buildings*, *78*, 113–122.

Zhang, R., & Chiong, R. (2016). Solving the energy-efficient job shop scheduling problem: A multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption. *Journal of Cleaner Production*, *112*, 3361–3375.

Zhang, M., Jiao, Z., Ran, L., & Zhang, Y. (2023). Optimal energy and reserve scheduling in a renewable-dominant power system. *Omega*, *118*, Article 102848.