# Capacity planning with uncertain endogenous technology learning

Tushar Rathi and Qi Zhang [*]

*Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, MN 55455, USA*

**Abstract**

Optimal capacity expansion requires complex decision-making, often influenced by technology learning, which represents the reduction in expansion cost due to factors such as cumulative installed capacity. However, having perfect foresight over the technology cost reduction is highly unlikely. In this work, we develop a multistage stochastic programming framework to model capacity planning problems with endogenous uncertainty in technology learning. To assess the benefit of the proposed framework over deterministic optimization, we apply a shrinking-horizon approach to compute the value of stochastic solution. Further, a decomposition scheme based on column generation is developed to solve large instances. Results from our computational experiments indicate substantial potential cost savings and the effectiveness of the proposed decomposition algorithm in solving instances with large numbers of scenarios. Lastly, a power capacity planning case study is presented, highlighting the stochastic optimization's ability to anticipate significantly different expansion and production decisions in low- and high-learning scenarios.

**Keywords:** multistage stochastic programming, endogenous uncertainty, technology learning, value of stochastic solution, column generation

# 1 Introduction

Over the past few decades, the unfavorable shift in global climatic conditions has driven us to focus on renewable technology development to lower carbon emissions. The increasing energy demand has further aggravated the need to look for alternatives to traditional fossil-fuel-based energy sources. However, in addition to developing new technology, making them economical as fast as possible remains a challenging task. In general, the cost of a technology is a function of several interrelated factors, including pricing and the number of competitors, government regulations and policies, the scale of production, and demand, to name a few. The reduction in the cost of a new technology due to the above-mentioned factors is often termed technology learning.

---

[*]Corresponding author (qizh@umn.edu)

Of all the stated, the scale of production constitutes a major driving force for cost reduction in new technologies. One of the first reported instances of learning effect was in the aircraft industry (Wright, 1936), where the production cost of an aircraft was found to decrease with the quantity produced. Additionally, improved efficiency of workers with repetitive tasks has been extensively studied in different contexts, which in almost all cases leads to a reduction in cost/execution time (Wright, 1936; Laffel et al., 1992; Sturm, 1999). The type of learning where improved efficiency is observed solely by performing the same task repeatedly or due to a scale-up in production falls under the category of learning-by-doing.

The reduction in cost as a function of installed capacity is often expressed using learning curves, a common way of expressing technology learning. Lieberman (1984) discusses the concept of a learning curve and cost-affecting factors in chemical processing industries. Further, Daugaard et al. (2015) determine the correlation between the size/cost of biorefineries and installed capacity based on different learning curve models. Besides, learning curves have been used as a tool to estimate the time for a new technology to become cost-competitive. Rubin et al. (2007) utilize learning curves for cost projection of power plants with carbon capture. An assessment of solar power cost based on the extrapolation of the learning curve is presented in Van der Zwaan and Rabl (2003). Recently, a hybrid approach for estimating the cost of an $N^{th}$-of-a-kind plant (Rubin, 2019) was utilized for estimating future cost projections of $CO_2$ mineralization plants that can potentially help decarbonize the cement industry (Strunge et al., 2022). A detailed review of learning curve models and potential areas of application can be found in Anzanello and Fogliatto (2011).

A less considered aspect is the utilization of learning curves to make optimal capacity expansion decisions. Heuberger et al. (2017) account for endogenous technology learning in a power capacity expansion problem. Chen et al. (2017) present a dynamic programming framework that integrates learning curves for making decisions to advance low-carbon fuels. Aliabadi (2020) incorporates endogenous technology learning in a model that outputs optimal capacities and deployment times for coal-based power plants equipped with carbon capture, utilization, and storage (CCUS) technology. Recently, Bakker et al. (2021) investigate the effect of incorporating endogenous learning in determining decisions regarding the plugging and abandonment of oil and gas wells.

Most of the literature on optimization concerning learning curves assumes that they can be constructed deterministically. Historical data is generally used to construct learning curves, which, in general, could be a useful technique if the data is readily available as well as reliable. However, the sheer unavailability of data, the dependence of the learning on the decisions made in real-time, or the influence of other external factors can make predicting the learning curve a challenging task. For example, an unexpected technological breakthrough (exogenous technological change) could affect how the learning curve will develop. Moreover, government safety regulations, especially in the context of chemical plants, can hinder the decrease in plant expansion costs. A discussion of factors affecting learning rates can be found in Rout et al. (2009). The unpredictability of such factors at the time of decision-making introduces uncertainty in the learning curve.

It has been observed that the policy decisions made with models that consider learning curves are often upward biased (anticipating higher learning rate) if the external factors are neglected (Nordhaus, 2014). The upward bias could lead to sub-optimal or infeasible solutions that could be detrimental, especially for long-term planning projects. Moreover, uncertainty in learning rates has been accounted for, if at all, using methods such as Monte Carlo simulation (Kim et al., 2012). Even though such methods can provide valuable insights, their inability to ensure non-anticipativity

demands a more rigorous optimization framework. For this reason, we explore the feasibility of stochastic programming (Birge and Louveaux, 2011) in incorporating uncertain learning curves for multiperiod capacity expansion planning.

In stochastic programming, uncertainty is usually classified as exogenous or endogenous. The uncertainty not affected by decisions is termed exogenous, whereas the uncertainty affected by decisions is termed endogenous (Jonsbråten et al., 1998). Endogenous uncertainty is further classified as type-1 and type-2. Type-1 uncertainty arises when decisions alter the probability distribution of the uncertain parameters (Ahmed, 2000; Peeta et al., 2010; Hellemo et al., 2018), whereas type-2 uncertainty affects the timing of the realization of the uncertain parameters (Goel and Grossmann, 2006). For example, the size of an oil reserve and the gas production rate are only revealed after drilling the reserve (Goel and Grossmann, 2004; Gupta and Grossmann, 2014). Similarly, in a pharmaceutical planning problem, the outcome of a clinical trial resolves only when the potential drug is subjected to the trial (Colvin and Maravelias, 2008, 2010). Likewise, in a capacity expansion problem with an uncertain learning curve, the expansion cost realizes only when the capacity is actually increased; thus, the uncertainty in expansion cost classifies as type-2 endogenous. Recently, Zhang and Feng (2020) further refined the type-2 classification by differentiating between decision-dependent materialization (type-2a) and observation (type-2b) of uncertain parameters. Apap and Grossmann (2017) provide a comprehensive review on the application of stochastic programming to problems with exogenous and type-2 endogenous uncertainty.

The main contributions of this work, along with the organization of the paper, are summarized as follows:

1. In Section 2, we consider endogenous technology learning and develop a mixed-integer linear programming (MILP) capacity planning formulation for a general process network comprising a set of processes/technologies and resources. In Section 3, we propose a stochastic optimization framework to account for type-2 endogenous uncertainty in technology learning.

2. An algorithm to compute the value of stochastic solution for multistage problems with endogenous uncertainty is developed in Section 4. The algorithm is crucial for quantifying the value of a stochastic optimization framework over a deterministic approach.

3. In Section 5, we exploit the proposed capacity planning formulation structure to devise an efficient decomposition scheme based on column generation. The proposed scheme leads to better feasible solutions and improved computation times.

4. We conduct extensive computational experiments on randomly generated instances of varying sizes (in terms of the number of scenarios and the length of the planning horizon) in Section 6. The effectiveness of the proposed framework and the decomposition scheme is showcased through the resulting values of stochastic solutions and improved computational statistics, respectively.

5. In Section 7, we demonstrate the practicability of the proposed framework through a case study on power capacity planning. In particular, we highlight the differences in expansion and production decisions for low- and high-learning scenarios, emphasizing the capability

of the proposed framework to produce sound decisions while ensuring non-anticipativity. Lastly, we conclude with some final remarks in Section 8.

# 2   Deterministic model

To capture the interconnectivity of technologies, model their simultaneous availability to satisfy product demand, and optimize their selection for capacity expansion and operations, we consider a general process network comprising process and resource nodes as illustrated in Figure 1. Processes and resources are denoted by square and circular nodes, respectively. The arcs in the network denote the directed flow of resources. Process nodes can represent chemical and manufacturing processes or, generally, technologies. Resource $j \in \mathcal{J}$ produced by a process $k \in \mathcal{K}$ can either serve as an input resource (denoted by black arrows) to process $k' \in \mathcal{K}\backslash\{k\}$ or discharged from the process network (denoted by red arrows). Moreover, a resource can also be purchased or made available from an outside source (denoted by blue arrows).
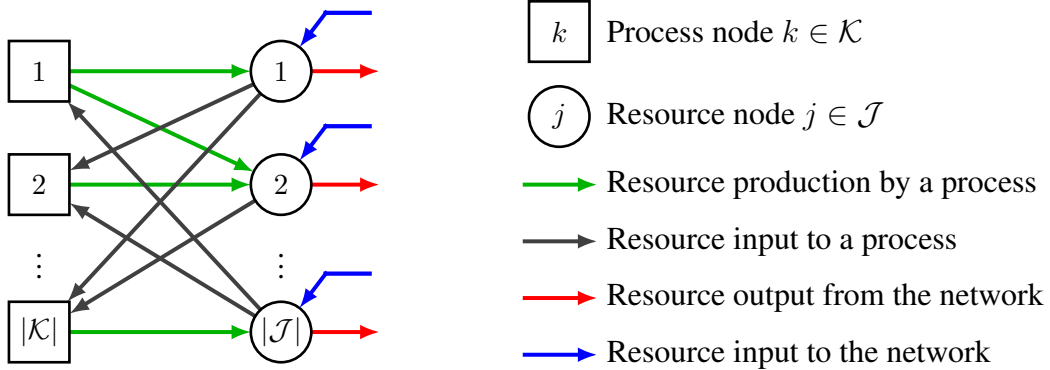


**Figure 1:** A general process network comprising a set of processes $\mathcal{K}$ (square nodes) and a set of resources $\mathcal{J}$ (circular nodes). Production and consumption of resources by processes are denoted by green and black arrows, respectively. The flow of resources to and from the network are denoted by blue and red arrows, respectively.

We assume that some processes may exhibit the phenomenon of technology learning. Learning curves are a pictorial depiction of the concept of endogenous technology learning representing a reduction in capital costs due to various factors, including but not limited to scale-up and R&D investments. In this paper, we focus on univariate learning curves based on the concept of learning-by-doing, which, in a production setting, represents cost reduction due to capacity expansion. Anzanello and Fogliatto (2011) provide a detailed comparison of various univariate models and discussion of multivariate models. Amongst the most studied univariate models for cost prediction is the log-linear model. Figure 2 shows a typical deterministic log-linear learning curve illustrating the reduction in unit expansion cost with cumulative installed capacity. The learning rate is usually significant during the initial development stages of a technology or a new plant setup phase, usually with low capacities. As the technology matures (or the plant is scaled-up), the learning rate decreases, eventually plateauing towards the end of its lifetime.
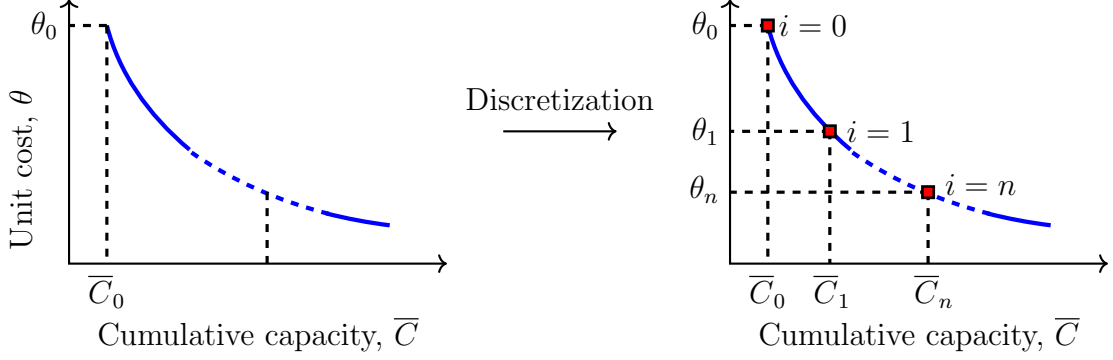
**Figure 2:** A log-linear learning curve and its discretization.

A mathematical representation of a log-linear model is as follows:

$$f(\overline{C}) = \theta = \theta_0 \left( \frac{\overline{C}}{\overline{C}_0} \right)^l$$

$$\text{or } f(\overline{C}_i) = \theta_i = \theta_0 \left( \frac{\overline{C}_i}{\overline{C}_0} \right)^l \text{ (discretized version)}$$

where $\overline{C}_0$ and $\theta_0$ denote the initial installed capacity and initial unit expansion cost, respectively. The slope of the learning curve is controlled by the parameter $l \leq 0$. The unit expansion cost after the cumulative installed capacity increases to $\overline{C}$ is denoted by $\theta$. To maintain tractability, we allow capacity to take only a finite number of values belonging to the set $\mathcal{I}$; thus, discretizing the learning curve as illustrated in Figure 2. Here, it is important to note that the unit expansion cost ($\theta_i$) materializes only when we actually expand the cumulative capacity to $\overline{C}_i$. Therefore, the technology learning curve is endogenous.

The goal is to determine optimal capacity expansion decisions in a long-term planning problem as the resource demand grows with time. Additionally, the model considers operational constraints within each time period $t$; thus, allowing determining optimal operational decisions based on factors including each process's available capacity, the demand of resources, up-time/down-time of units, and so on. To accomplish this goal, we start with proposing a deterministic mixed-integer linear programming (MILP) model for capacity expansion with technology learning. In the deterministic model, we assume that there is no uncertainty, i.e., the learning curve for each process $k \in \mathcal{K}$ is known precisely. Also, we use the terms "process" and "technology" interchangeably from here onwards.

## 2.1 Capacity expansion constraints

Based on the process network in Figure 1, we define binary variable $x_{kit}$ that equals 1 if a process $k$ undergoes capacity expansion to (at least) the permissible point $i \in \mathcal{I}$ in time period $t \in \mathcal{T}$. We further define the variables $C_{kt}$ and $\Delta_{kt}$ representing the cumulative installed capacity and additional capacity installed of a process $k$ in time period $t$, respectively. Then, the following constraints control the timing and extent of capacity expansion for each process in the network:

$$C_{k0} = \overline{C}_{k0} \qquad\qquad \forall k \in \mathcal{K} \quad \text{(1a)}$$

$$C_{kt} = C_{k,t-1} + \Delta_{kt} \qquad\qquad \forall k \in \mathcal{K}, t \in \mathcal{T} \quad \text{(1b)}$$

$$\Delta_{kt} = \sum_{i \in \mathcal{I}_k} x_{kit} \overline{\Delta}_{ki} \qquad\qquad \forall k \in \mathcal{K}, t \in \mathcal{T} \quad \text{(1c)}$$

$$\Delta_{kt} \leq b_{kt} \qquad\qquad \forall k \in \mathcal{K}, t \in \mathcal{T} \quad \text{(1d)}$$

$$x_{kit} \leq \sum_{\tau=1}^{t} x_{k,i-1,\tau} \qquad\qquad \forall k \in \mathcal{K}, i \in \mathcal{I}_k \backslash \{1\}, t \in \mathcal{T} \quad \text{(1e)}$$

$$\sum_{\tau=1}^{t} x_{ki\tau} \leq 1 \qquad\qquad \forall k \in \mathcal{K}, i \in \mathcal{I}_k, t \in \mathcal{T} \quad \text{(1f)}$$

$$x_{kit} \in \{0,1\} \qquad\qquad \forall k \in \mathcal{K}, i \in \mathcal{I}_k, t \in \mathcal{T} \quad \text{(1g)}$$

$$C_{kt}, \Delta_{kt} \geq 0 \qquad\qquad \forall k \in \mathcal{K}, t \in \mathcal{T} \quad \text{(1h)}$$

where $\overline{C}_{k0}$ denotes the initial installed capacity of process $k$. The incremental capacity for process $k$ from point $i - 1$ to $i$ is denoted by $\overline{\Delta}_{ki}$. Constraints (1a)-(1c) together represent the capacity balance. Constraints (1d) limit the capacity expansion of a process $k$ by the available budget $b_{kt}$ in time period $t$. Constraints (1e) ensure that we move in the positive direction on the learning curve in a sequential fashion, i.e., we can only install additional capacity corresponding to point $i$ if we have already installed the additional capacity corresponding to point $i - 1$. That being said, it does not restrict installing additional capacities corresponding to both point $i - 1$ and $i$ in the same time period $t$, which can be alternatively interpreted as jumping directly to the point $i$ from a lower capacity point. Constraints (1f) imply that investment at any point $i \in \mathcal{I}$ cannot be made more than once in any time period.

## 2.2 Operational constraints

In addition to modeling the capacity expansion decisions, we further solve an operational problem at the scheduling level. We select a representative scheduling horizon $\mathcal{H}_t$ (e.g., a representative day) for each time period $t$ to facilitate the modeling of time-varying operation, which is particularly important in systems involving intermittent renewable energy, and we assume that it is repeated $n_t$ times in time period $t$. As illustrated in Figure 3, each $\mathcal{H}_t$ is of length $H_t$ (e.g., $H_t$ hours). Different time periods are not restricted to having scheduling horizons of the same length.
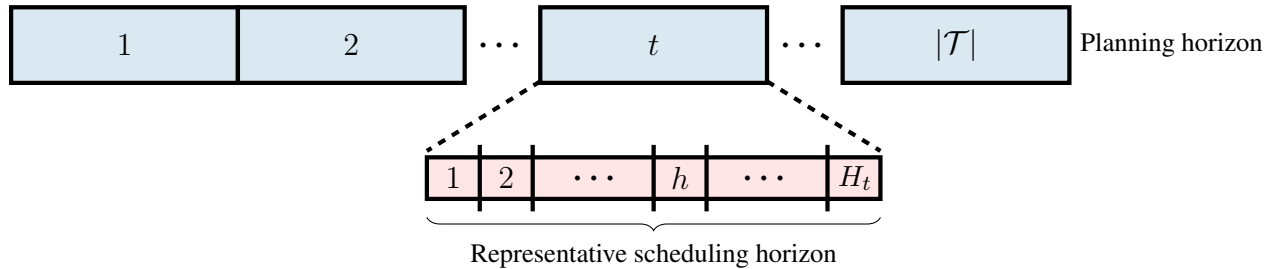


**Figure 3:** Multiscale time representation, which divides the planning horizon into a set of time periods, $\mathcal{T}$, with each time period $t \in \mathcal{T}$ having a representative scheduling horizon $\mathcal{H}_t$ of length $H_t$.

We define variables $Q_{jth}$ and $P_{kth}$ to represent the inventory of resource $j$ and the amount of the reference resource produced or consumed by process $k$ in interval $h$ of time period $t$, respectively.

6

Further, we define a variable $V_{jth}$ that denotes the influx of resource $j$ in interval $h$ of time period $t$. It can be used to mitigate the resource shortage, purchase a resource at a lower price, make available renewable resources such as biomass, or denote the unmet demand. Discrete decisions involving the number of operating units (or units started/shut down) for a process $k$ in interval $h$ of time period $t$ is represented through variables $y_{kth}$. The demand of resource $j$ in interval $h$ of time period $t$ is denoted by $d_{jth}$, respectively. Parameter $\rho_{kjt}$ serves as a conversion factor to deduce the amount of resource $j$ produced or consumed by process $k$ based on its reference resource. The fraction of cumulative capacity of process $k$ available for utilization in interval $h$ of time period $t$ is denoted by $\eta_{kth}$. It is particularly useful to model processes with time-varying capacities, e.g., solar- and wind-based power generation. The operational decisions can then be modeled using the following constraints:

$$Q_{jth} = Q_{jt,h-1} + \sum_{k \in \mathcal{K}} \rho_{kjt} P_{kth} + V_{jth} - d_{jth} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T}, h \in \mathcal{H}_t \quad \text{(2a)}$$

$$Q_{jtH_t} \geq Q_{jt0} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad \text{(2b)}$$

$$P_{kth} \leq \eta_{kth} C_{kt} \qquad \forall k \in \mathcal{K}, t \in \mathcal{T}, h \in \mathcal{H}_t \quad \text{(2c)}$$

$$g_{kth}(C_{kt}, Q_{jth}, P_{kth}, y_{kth}, ...) \leq 0 \qquad \forall k \in \mathcal{K}, j \in \mathcal{J}, t \in \mathcal{T}, h \in \mathcal{H}_t \quad \text{(2d)}$$

$$0 \leq V_{jth} \leq V_{jth}^{\max} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T}, h \in \mathcal{H}_t \quad \text{(2e)}$$

$$Q_{jth} \geq 0 \qquad \forall j \in \mathcal{J}, t \in \mathcal{T}, h \in \mathcal{H}_t \quad \text{(2f)}$$

$$P_{kth} \geq 0 \qquad \forall k \in \mathcal{K}, t \in \mathcal{T}, h \in \mathcal{H}_t \quad \text{(2g)}$$

$$y_{kth} \in \mathbb{Z} \qquad \forall k \in \mathcal{K}, t \in \mathcal{T}, h \in \mathcal{H}_t \quad \text{(2h)}$$

where constraints (2a) and (2b) denote the inventory balance. Constraints (2c) enforce time-varying availability of the installed capacity for each process $k$. Constraints (2d) represent the remaining relevant operational constraints. For example, we can constrain inventory by limiting storage capacity, model mode-based operations for each process $k$, and model startup and shut down of operations for each process, to name a few. Constraints (2e)-(2h) define bounds on the operational variables.

## 2.3 Objective function

The capital expenditure (CAPEX) comprises the scaling-up cost for each process $k$ incurred in each time period $t$ and is represented as follows:

$$C^{\text{CPX}} = \sum_{t \in \mathcal{T}} \alpha_t \left[ \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}_k} \left( \int_{\Phi_{k,i-1}}^{\Phi_{ki}} f_k(\Phi_k) \mathrm{d}\Phi_k \right) x_{kit} \right] \qquad (3)$$

where $\alpha_t$ denotes the discount factor for time period $t$. The learning curve for process $k$ is encoded in the model as $f_k(\Phi_k)$ and $\Phi_{ki} := \sum_{i'=1}^{i} \overline{\Delta}_{ki'}$. Note that we make no assumptions on the form of the learning curve since the integral term (expansion cost on increasing capacity from point $i-1$ to $i$) is a parameter that can be pre-calculated in the pre-modeling phase irrespective of the form of the learning curve (Figure 4).

The net operating cost (OPEX) can be represented as follows:

$$C^{\text{OPX}} = \sum_{t \in \mathcal{T}} \alpha_t n_t \sum_{h \in \mathcal{H}_t} \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} \left( \beta_{kjth} \rho_{kjt} P_{kth} + u_{kjth}(C_{kt}, Q_{jth}, P_{kth}, y_{kth}, ...) \right) \qquad (4)$$
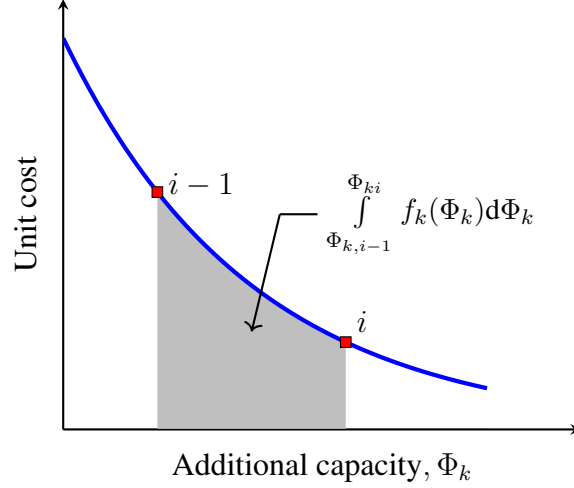
**Figure 4:** The cost of expansion for process $k$ from point $i-1$ to $i$ is given by the area under the learning curve. Note that we consider learning as a function of additional capacity installed over time instead of the cumulative capacity. We do so because the net capacity may decrease if we allow the capacities to retire after a certain duration, whereas the additional capacity installed can only increase.

where $\beta_{kjth}$ represents the unit production cost of resource $j$ by process $k$ in interval $h$ of time period $t$. The first term in equation (4) denotes the cost of production, whereas the second term, $u(\cdot)$, captures the operating cost of specific modes of operation, utilizing storage, purchasing and discharging resources, tax on emissions, etc.

The net present cost, $C^{\text{NET}}$, is simply the sum of capital and operating expenditures:

$$C^{\text{NET}} = C^{\text{CPX}} + C^{\text{OPX}} \tag{5}$$

where the objective is to minimize the net cost for the entire planning horizon.

The final deterministic capacity expansion problem with technology learning can be summarized as:

$$\text{minimize} \quad C^{\text{NET}} \tag{FS$_{\text{DM}}$}$$
$$\text{subject to} \quad \text{Eqs. (1a) - (5).}$$

# 3 Stochastic programming model

As stated in Section 1, there can be uncertainty in the learning curve due to lack of historical data, safety regulations and policies, technological breakthroughs, and other external factors. Moreover, in a realistic setting, the assumption of a univariate learning curve rarely holds. Therefore, we need to account for uncertainty in the learning curve, i.e., we need to consider different scenarios regarding how the learning curve might unfold. Figure 5 shows an example of how a learning curve may take different paths as additional capacity is installed. Again, we consider only a finite number of expansion points. Also, in this work, we focus on the uncertainty in the learning curve of expansion cost and hence neglect other potential sources of uncertainty such as demand.

Each possible expansion point acts as a source of uncertainty (dashed boxes in Figure 5: represented as $i$), and the uncertainty in cost at permissible point $i$ resolves only when we increase the
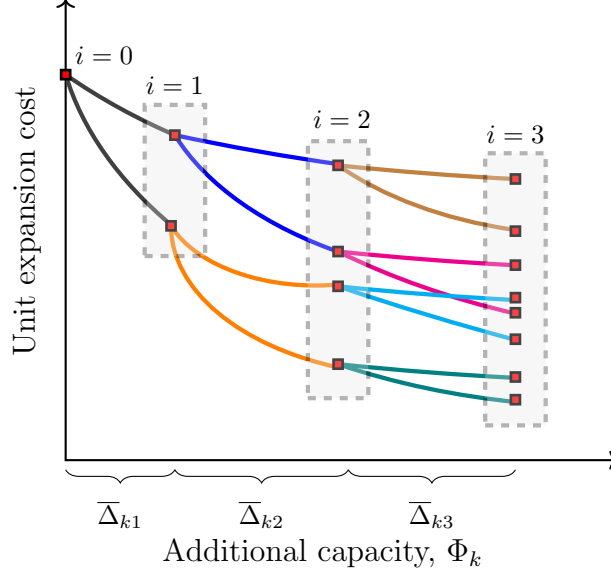
**Figure 5:** Example of an uncertain learning curve. Each discrete expansion point $i$ acts as source of uncertainty. In this case, we have two, four, and eight possible unit expansion costs at $i = 1, 2$, and $3$, respectively. Note that this is not a scenario tree but simply an uncertainty representation; however, it does help in creating scenarios for the overall problem.

capacity to point $i$; therefore, it classifies as a type-2 endogenous uncertainty. Furthermore, as the uncertainty resolves, our anticipation of the learning curve from that point onwards changes too. In other words, any point of realization (red markers in Figure 5) can be thought of as a new starting point. Thus, a new learning curve unfolds from that point onwards with a different underlying uncertainty. This process continues until we reach the limit of capacity addition or the planning horizon.

We utilize the concept of stochastic programming to extend the deterministic model of Section 2 to account for uncertainty in the technology learning curves. Since our planning horizon spans multiple time periods, we specifically formulate a multistage stochastic programming model. Further, the presence of binary capacity expansion decisions (and likely integer operational decisions) leads to an MILP model. The specifics of how the deterministic model is modified to account for uncertainty are discussed in the subsequent subsections.

## 3.1 Scenario feasibility constraints

The core idea in stochastic programming is to account for uncertainties through scenarios, which in our case are given in the form of possible learning curves (or a combination of them in case of multiple uncertain technologies/processes). Based on this idea, the stochastic model can be interpreted as a collection of deterministic models, one for each possible scenario $s \in \mathcal{S}$. Therefore,

we extend the capacity expansion and operational constraints to each possible scenario as follows:

$$
\left.
\begin{aligned}
&C_{k0s} = \overline{C}_{k0} && \forall k \in \mathcal{K} \\
&C_{kts} = C_{k,t-1,s} + \Delta_{kts} && \forall k \in \mathcal{K}, t \in \mathcal{T} \\
&\Delta_{kts} = \sum_{i \in \mathcal{I}_k} x_{kits}\overline{\Delta}_{ki} && \forall k \in \mathcal{K}, t \in \mathcal{T} \\
&\Delta_{kts} \leq b_{kt} && \forall k \in \mathcal{K}, t \in \mathcal{T} \\
&x_{kits} \leq \sum_{\tau=1}^{t} x_{k,i-1,\tau s} && \forall k \in \mathcal{K}, i \in \mathcal{I}_k \backslash \{1\}, t \in \mathcal{T} \\
&\sum_{\tau=1}^{t} x_{ki\tau s} \leq 1 && \forall k \in \mathcal{K}, i \in \mathcal{I}_k, t \in \mathcal{T} \\
&Q_{jths} = Q_{jt,h-1,s} + \sum_{k \in \mathcal{K}} \rho_{kjt} P_{kths} + V_{jths} - d_{jth} && \forall j \in \mathcal{J}, t \in \mathcal{T}, h \in \mathcal{H}_t \\
&Q_{jtH_ts} \geq Q_{jt0s} && \forall j \in \mathcal{J}, t \in \mathcal{T} \\
&P_{kths} \leq \eta_{kth} C_{kts} && \forall k \in \mathcal{K}, t \in \mathcal{T}, h \in \mathcal{H}_t \\
&g_{kths}(C_{kts}, Q_{jths}, P_{kths}, y_{kths}, ...) \leq 0 && \forall k \in \mathcal{K}, j \in \mathcal{J}, t \in \mathcal{T}, h \in \mathcal{H}_t \\
&0 \leq V_{jths} \leq V_{jth}^{\max} && \forall j \in \mathcal{J}, t \in \mathcal{T}, h \in \mathcal{H}_t \\
&Q_{jths} \geq 0 && \forall j \in \mathcal{J}, t \in \mathcal{T}, h \in \mathcal{H}_t \\
&P_{kths} \geq 0, y_{kths} \in \mathbb{Z} && \forall k \in \mathcal{K}, t \in \mathcal{T}, h \in \mathcal{H}_t \\
&C_{kts}, \Delta_{kts} \geq 0 && \forall k \in \mathcal{K}, t \in \mathcal{T} \\
&x_{kits} \in \{0,1\} && \forall k \in \mathcal{K}, i \in \mathcal{I}_k, t \in \mathcal{T}
\end{aligned}
\right\} \forall s \in \mathcal{S}
\tag{6}
$$

where all decision variables have the same meaning as in the deterministic model; however, here, they are also indexed for each scenario $s \in \mathcal{S}$, ensuring the feasibility of eventual decisions for all the considered scenarios.

## 3.2 Non-anticipativity constraints

Now, ensuring the feasibility of decisions for all scenarios is usually not sufficient. A less intuitive (more so in the case of endogenous uncertainty) but equally important condition is to ensure equality of decisions for indistinguishable scenarios at all points in time of the planning horizon. Two scenarios $s$ and $s'$ are considered indistinguishable at time $t$ if no uncertain source that distinguishes the two scenarios has been resolved. To ensure $s$ and $s'$ have the exact same capacity expansion decisions implemented all the way from $t = 1$ to the beginning of time period $t + 1$, we need additional constraints. These constraints are termed non-anticipativity constraints (NACs) and link the capacity expansion decisions for the indistinguishable scenarios. The following representation of conditional NACs using disjunctions and logic constraints is adapted from Goel and Grossmann (2006).

$$
x_{ki1s} = x_{ki1s'} \qquad\qquad \forall s, s' \in \mathcal{S}, s \neq s', k \in \mathcal{K}, i \in \mathcal{I}_k \tag{7a}
$$

$$\begin{bmatrix} Z_t^{s,s'} \\ x_{ki,t+1,s} = x_{ki,t+1,s'} \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k \end{bmatrix} \vee \begin{bmatrix} \neg Z_t^{s,s'} \end{bmatrix} \qquad \forall s, s' \in \mathcal{S}, s \neq s', t \in \mathcal{T} \backslash \{T\} \quad (7b)$$

$$Z_t^{s,s'} \iff \bigwedge_{(r,i) \in \mathcal{D}(s,s')} \begin{bmatrix} \bigwedge_{\tau=1}^{t} (\neg x_{ri\tau s}) \end{bmatrix} \qquad \forall s, s' \in \mathcal{S}, s \neq s', t \in \mathcal{T} \backslash \{T\} \quad (7c)$$

$$Z_t^{s,s'} \iff \bigwedge_{(r,i) \in \mathcal{D}(s,s')} \begin{bmatrix} \bigwedge_{\tau=1}^{t} (\neg x_{ri\tau s'}) \end{bmatrix} \qquad \forall s, s' \in \mathcal{S}, s \neq s', t \in \mathcal{T} \backslash \{T\} \quad (7d)$$

$$Z_t^{s,s'} \in \{\texttt{true}, \texttt{false}\} \qquad \forall s, s' \in \mathcal{S}, s \neq s', t \in \mathcal{T} \backslash \{T\} \quad (7e)$$

Since no decision has been implemented at the start of the first time period, all scenarios are the indistinguishable at that point in time. This condition is enforced in constraints (7a) by the equality of capacity expansion decisions at the very beginning of the planning horizon. Disjunctions (7b) impose NACs for each scenario pair $(s, s')$ provided that the Boolean variable $Z_t^{s,s'}$ is $\texttt{true}$. Constraints (7c) and (7d) relate the Boolean variable $Z_t^{s,s'}$ with the capacity expansion decisions for the technologies with uncertain learning curves, denoted by set $\mathcal{R}$. Specifically, if $\theta_{ris}$ is the uncertainty realization of source $(r, i)$ in scenario $s$, and $\mathcal{D}(s, s') = \{(r, i) | r \in \mathcal{R}, i \in \mathcal{I}_r, \theta_{ris} \neq \theta_{ris'}\}$ is the set of sources of endogenous uncertainty that distinguish scenario $s$ from $s'$, then $Z_t^{s,s'}$ is $\texttt{true}$ if at the end of time period $t$, the uncertainty has not realized in any of the uncertain parameters that belong to the set $\mathcal{D}(s, s')$. It is important to note here that constraints (7a)-(7e) are a mathematical representation of the conditional scenario tree, which is a unique characteristic of stochastic optimization problems with (type-2) endogenous uncertainty.

Now, clearly the set $\mathcal{P} = \{(s, s') | s, s' \in \mathcal{S}, s \neq s'\}$ can become huge as the number of scenarios grows. Consequently, the number of NACs (7a)-(7e) can become exponentially large. However, usually, a large fraction of these NACs are redundant. To identify the redundant scenario pairs, we utilize a polynomial-time exact algorithm discussed in Hooshmand and MirHassani (2016) designed for the case of pure endogenous uncertainty with an arbitrary scenario set. Since model reduction is not the focus of this paper, we do not delve into the details of the NAC reduction method. We refer the reader to Hooshmand and MirHassani (2016) for the details of the algorithm.

Let $\mathcal{P}'$ denote the reduced set of scenario pairs, then the reduced equivalent of NACs (7a)-(7e) is as follows:

$$x_{ki1s} = x_{ki1,s+1} \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k, s \in \mathcal{S}, s < |\mathcal{S}| \qquad (8a)$$

$$\begin{bmatrix} Z_t^{s,s'} \\ x_{ki,t+1,s} = x_{ki,t+1,s'} \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k \end{bmatrix} \vee \begin{bmatrix} \neg Z_t^{s,s'} \end{bmatrix} \quad \forall (s, s') \in \mathcal{P}', t \in \mathcal{T} \backslash \{T\} \qquad (8b)$$

$$Z_t^{s,s'} \iff \bigwedge_{(r,i) \in \mathcal{D}(s,s')} \begin{bmatrix} \bigwedge_{\tau=1}^{t} (\neg x_{ri\tau s}) \end{bmatrix} \quad \forall (s, s') \in \mathcal{P}', t \in \mathcal{T} \backslash \{T\} \qquad (8c)$$

$$Z_t^{s,s'} \in \{\texttt{true}, \texttt{false}\} \quad \forall (s, s') \in \mathcal{P}', t \in \mathcal{T} \backslash \{T\} \qquad (8d)$$

The above modifications preserve the optimal solution of the original formulation. The reformulation of constraints (8b) and (8c) to linear constraints is given in Sections A and B of the supplementary material, respectively.

## 3.3 Objective function

We define CAPEX for each scenario based on the corresponding capacity expansion decisions as follows:

$$C_s^{\text{CPX}} = \sum_{t \in \mathcal{T}} \alpha_t \left[ \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}_k} \left( \int_{\Phi_{k,i-1}}^{\Phi_{ki}} f_{ks}(\Phi_k) \mathrm{d}\Phi_k \right) x_{kits} \right] \qquad \forall s \in \mathcal{S} \quad (9)$$

Similarly, the operational decisions in each scenario determine the OPEX as follows:

$$C_s^{\text{OPX}} = \sum_{t \in \mathcal{T}} \alpha_t n_t \sum_{h \in \mathcal{H}_t} \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} \left( \beta_{kjth} \rho_{kjt} P_{kths} + u_{kjth}(C_{kts}, Q_{jths}, P_{kths}, y_{kths}, ...) \right) \qquad \forall s \in \mathcal{S}$$

$$(10)$$

For each scenario $s$, the net cost equals the sum of the corresponding CAPEX and OPEX:

$$C_s^{\text{NET}} = C_s^{\text{CPX}} + C_s^{\text{OPX}} \qquad \forall s \in \mathcal{S} \quad (11)$$

For the capacity expansion problem with uncertain endogenous technology learning, the objective is to minimize the expected net cost over the entire planning horizon; thus, the overall stochastic optimization problem can be summarized as follows:

$$\text{minimize} \quad \sum_{s \in \mathcal{S}} p_s C_s^{\text{NET}} \qquad \qquad \text{(FS}_{\text{SP}})$$

subject to Eqs. (6), (8a) - (8d), (9) - (11)

where $p_s$ denotes the probability of scenario $s$.

# 4 Value of stochastic solution for multistage problems with endogenous uncertainty

The value of stochastic solution (VSS) provides a quantitative measure of the benefits yielded from accounting for uncertainty in parameters instead of modeling with their expected values, as illustrated by equation (12). It indicates whether it is worth investing in formulating and solving a stochastic program (SP) instead of a deterministic expected value problem (EVP). For the most part, VSS has been defined and implemented for two-stage stochastic optimization problems with exogenous uncertainty (Birge and Louveaux, 2011). Lately, the concept has been extended to multistage stochastic optimization problems with exogenous uncertainty (Escudero et al., 2007; Maggioni et al., 2014; Zhang et al., 2018); however, it remains sparsely utilized. To the best of our knowledge, the algorithm for calculating VSS has not been formally developed for stochastic optimization problems with endogenous uncertainty.

$$\text{VSS} = \mathbb{E}[Z^{\text{EVP}}] - Z^{\text{SP}} \qquad (12)$$

We present a shrinking-horizon approach for solving the EVP at each node of the scenario tree for multistage problems. Unlike the classical exogenous case, the main challenge in determining

the VSS for the endogenous case lies in constructing the conditional scenario tree based on the investment decisions that control which and when an uncertain source resolves.

To develop the algorithm, we use a notation independent of the notation used in the rest of the paper. Following are the sets and parameters utilized in this section: Set of uncertainty sources is denoted by $\mathcal{I}$. For the sake of brevity, we assume that each uncertainty source has only one uncertain parameter associated with it; however, the algorithm works for sources with multiple uncertainties as well. Realizations of the uncertain parameter associated with source $i$ form the set $\mathcal{R}_i$. Time periods are represented by the set $\mathcal{T}$. The last time period is denoted by parameter $T$. The scenario tree is characterized by 'layers' that form the set $\mathcal{L}$, where a layer $l \in \mathcal{L}$ has a set of nodes represented by $\mathcal{N}_l$. It is important to note that the cardinality of the set $\mathcal{L}$ is always one more than the set of time periods $\mathcal{T}$, i.e., $|\mathcal{L}| = T + 1$. A node is represented by the pair $(l, n)$ indicating the layer $l$ it resides in and its specific label $n$. The set of children nodes of a node $(l, n)$ is denoted by $\mathcal{C}_{ln}$, and the parent node of node $(l, n)$ resides in layer $l - 1$ and is denoted by $\bar{n}_{ln}$. The uncertainty sources resolved at node $(l, n)$ are represented by the set $\mathcal{I}'_{ln}$, whereas the set $\widetilde{\mathcal{I}}_{ln}$ comprises all uncertainty sources resolved prior to node $(l, n)$. Lastly, the values of all the uncertain parameters realized prior to node $(l, n)$ are contained in vector $\hat{\Theta}_{ln}$. In contrast, the expected values of those yet to be resolved comprises vector $\overline{\Theta}_{ln}$. Figure 6 illustrates the notation through a small example.



$\mathcal{N}_1 = \{1\}$    $\bar{n}_{33} = 2$    $\mathcal{I}'_{22} = \emptyset$    $\hat{\Theta}_{32} = (\hat{\theta}_1^L, \hat{\theta}_2^H)$
$\mathcal{N}_2 = \{1, 2\}$    $\mathcal{C}_{11} = \{(2,1), (2,2)\}$    $\widetilde{\mathcal{I}}_{32} = \{1, 2\}$    $\overline{\Theta}_{11} = (\mathbb{E}[\theta_1], \mathbb{E}[\theta_2])$
$\mathcal{N}_3 = \{1, 2, 3\}$    $\mathcal{C}_{22} = \{(3,3)\}$    $\hat{\Theta}_{21} = (\hat{\theta}_1^L)$    $\overline{\Theta}_{21} = \overline{\Theta}_{22} = \overline{\Theta}_{33} = (\mathbb{E}[\theta_2])$
$\bar{n}_{21} = 1$    $\mathcal{I}'_{11} = \{1\}$    $\hat{\Theta}_{22} = \hat{\Theta}_{33} = (\hat{\theta}_1^H)$
$\bar{n}_{31} = 1$    $\mathcal{I}'_{21} = \{2\}$    $\hat{\Theta}_{31} = (\hat{\theta}_1^L, \hat{\theta}_2^L)$
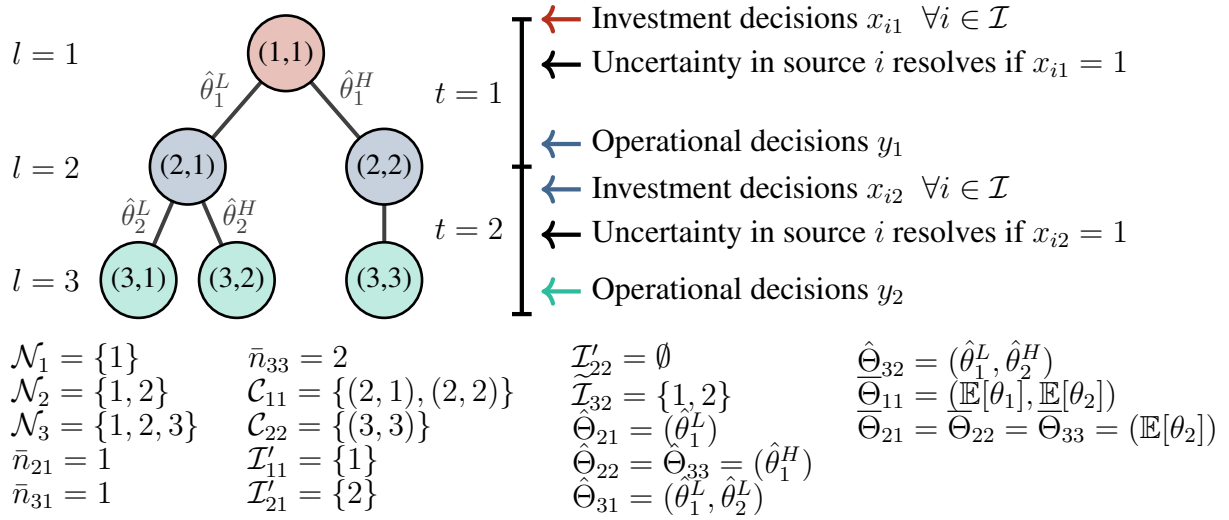
**Figure 6:** Illustrating notation using a problem with $\mathcal{I} = \{1, 2\}$, $\mathcal{R}_1 = \{\hat{\theta}_1^L, \hat{\theta}_1^H\}$, $\mathcal{R}_2 = \{\hat{\theta}_2^L, \hat{\theta}_2^H\}$, $\mathcal{T} = \{1, 2\}$. Note that $l$ denotes a layer in the scenario tree. The first time period ($t = 1$) spans from $l = 1$ to $l = 2$, and so on. Uncertainty in source $i = 1$ realizes after investment decisions in the first time period. If $\theta_1 = \hat{\theta}_1^L$ then the investment decisions in the second time period resolves the uncertainty in source $i = 2$; otherwise, source $i = 2$ remains unresolved.

The following formulation represents a general multistage stochastic program with endogenous uncertainty:

$$Z^{\text{MSP}} = \min_{x,y} \quad \sum_{s \in \mathcal{S}} p_s \sum_{t \in \mathcal{T}} f_t(x_{ts}, y_{ts}, \theta_s) \tag{MSP}$$

$$\text{subject to} \quad g_t(x_{[t]s}, y_{[t]s}, \theta_s) \geq 0 \quad \forall t \in \mathcal{T}, s \in \mathcal{S}$$

13

$$\text{NACs}$$
$$x_{ts} \in \{0,1\}^{|\mathcal{I}|} \quad \forall t \in \mathcal{T}, s \in \mathcal{S}$$

where vector $x_{ts}$ comprises binary investment variables for each uncertainty source $i \in \mathcal{I}$ in time period $t$ of scenario $s$, i.e., $x_{ts} = (x_{1ts}, ..., x_{|\mathcal{I}|ts})$. Uncertainty in source $i$ resolves at the beginning of time period $t$ if $x_{its}$ equals 1. Vector $y_{ts}$ comprises operational variables for time period $t$ of scenario $s$. Vectors $x_{[t]s} = (x_{1s}, ..., x_{ts})$ and $y_{[t]s} = (y_{1s}, ..., y_{ts})$ facilitate linking of decisions across time periods. Vector $\theta_s$ contains realizations of all uncertain parameters corresponding to scenario $s$, i.e., $\theta_s = (\theta_{1s}, ..., \theta_{|\mathcal{I}|s})$. The probability of a scenario $s$ is denoted by $p_s$. Objective $Z^{\text{MSP}}$ is a probability weighted sum of cost function $f_t$ of each time period $t$ in each scenario $s$. The feasible region of the problem is defined by NACs and the constraints on function $g_t$ for each time period $t$.

In a deterministic setting, an EVP is solved at each node of the conditional scenario tree. Due to endogenous uncertainty, modeling and solving the EVP requires us to dynamically construct the scenario tree based on the solution obtained at each node $(l, n)$. For this purpose, we define a series of sets and parameters that assist us in constructing the scenario tree.

The structure of the scenario tree corresponding to layer $l + 1$ is a function of the decisions made at the nodes in layer $l$. At node $(l, n)$ with $l \leq T$, we define a set comprising the uncertainty sources that get resolved after solving node $(l, n)$:

$$\mathcal{I}'_{ln} := \{i : i \in \mathcal{I}, x^*_{itn} = 1, t = l\} \tag{F-0}$$

As illustrated in Figure 6, no investment decisions are made at the leaf nodes since they correspond only to the recourse decisions for the last time period; thus, $\mathcal{I}'_{ln}$ is not defined for $l = T + 1$.

Further, we calculate the number of children nodes of a non-leaf node $(l, n)$ as follows:

$$\sigma_{ln} = \begin{cases} 1 & \text{if } \mathcal{I}'_{ln} = \emptyset \\ \prod_{i \in \mathcal{I}'_{ln}} |\mathcal{R}_i| & \text{otherwise} \end{cases} \tag{F-1}$$

where $\mathcal{R}_i$ denotes the set of realizations of uncertain parameters associated with source $i$.

Using $\sigma_{ln}$, the set containing children nodes of a non-leaf node $(l, n)$ can be defined as follows:

$$\mathcal{C}_{ln} := \begin{cases} \{(l+1, 1), ..., (l+1, \sigma_{ln})\}, & \text{if } n = 1 \\ \{(l+1, n') : n' \in \bigcup_{j=n''+1}^{j=n''+\sigma_{ln}} \{j\}, n'' = \sum_{m=1}^{n-1} \sigma_{lm}\}, & \text{otherwise} \end{cases} \tag{F-2}$$

Further, using the set $\mathcal{C}_{l-1,n}$, we define the ordered set $\mathcal{N}_l$ comprising nodes in layer $l$ of the scenario tree as follows:

$$\mathcal{N}_l := \begin{cases} \{1\}, & \text{if } l = 1 \\ \{n : (l, n) \in \bigcup_{n' \in \mathcal{N}_{l-1}} \mathcal{C}_{l-1,n'}\}, & \text{otherwise} \end{cases} \tag{F-3}$$

Now, let $\bar{n}_{ln}$ denote the parent node of node $(l, n)$ in parent layer, $l-1$. Note that when mapping parent to children nodes, we can simultaneously map the children to their parent node. Next, we

define the set of all the resolved uncertain sources prior to solving a node $(l,n)$:

$$\widetilde{\mathcal{I}}_{ln} := \begin{cases} \emptyset, & \text{if } l = 1 \\ \mathcal{I}'_{l-1,\bar{n}_{ln}} \cup \widetilde{\mathcal{I}}_{l-1,\bar{n}_{ln}}, & \text{if } l \in \mathcal{L}\backslash\{1\} \end{cases} \tag{F-4}$$

The probability of a node $(l,n)$ is defined as follows:

$$p_{ln} = \begin{cases} 1, & \text{if } l = 1 \\ p_{l-1,\bar{n}_{ln}}, & \text{if } l \in \mathcal{L}\backslash\{1\}, \mathcal{I}'_{l-1,\bar{n}_{ln}} = \emptyset \\ p_{l-1,\bar{n}_{ln}} \prod\limits_{i \in \mathcal{I}'_{l-1,\bar{n}_{ln}}} \varphi(\hat{\theta}^r_i), & \text{if } l \in \mathcal{L}\backslash\{1\}, \mathcal{I}'_{l-1,\bar{n}_{ln}} \neq \emptyset \end{cases} \tag{F-5}$$

where $\hat{\theta}^r_i$ is the realized value of uncertain parameter $i$, and $\varphi(\hat{\theta}^r_i)$ is the probability of realization of $\hat{\theta}^r_i$.

Returning to solving the EVP corresponding to the multistage problem (MSP), we present the following deterministic problem to be solved at node $(l,n)$ of the conditional scenario tree:

$$Z^{\text{DP}}_{ln} = \min_{x,y} \quad \sum_{t \in \mathcal{T}} f_t(x_t, y_t, \hat{\Theta}_{ln}, \overline{\Theta}_{ln}) \tag{DP$_{ln}$}$$

$$\text{subject to } g_t(x_{[t]}, y_{[t]}, \hat{\Theta}_{ln}, \overline{\Theta}_{ln}) \geq 0 \quad \forall t \in \mathcal{T} \tag{13a}$$
$$x_{it} = x^*_{it,l-1,\bar{n}_{ln}} \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, t \leq l-1 \tag{13b}$$
$$y_t = y^*_{t,l-1,\bar{n}_{ln}} \quad \forall t \in \mathcal{T}, t \leq l-2 \tag{13c}$$
$$x_t \in \{0,1\}^{|\mathcal{I}|} \quad \forall t \in \mathcal{T} \tag{13d}$$

where vector $\overline{\Theta}_{ln}$ comprises the expected value of uncertain parameters that are yet to be resolved, i.e., $\overline{\Theta}_{ln} = (E[\theta_{i_1}], E[\theta_{i_2}], ...) \quad \forall\{i_1, i_2, ...\} \in \mathcal{I}\backslash\widetilde{\mathcal{I}}_{ln}$. The expected value of uncertain parameter $\theta_i$, $\mathbb{E}[\theta_i] = \sum_{r \in \mathcal{R}_i} \hat{\theta}^r_i \varphi(\hat{\theta}^r_i)$. Similarly, vector $\hat{\Theta}_{ln}$ comprises the realized values of uncertain parameters already resolved, i.e., $\hat{\Theta}_{ln} = (\hat{\theta}_{i_1}, \hat{\theta}_{i_2}, ...) \quad \forall\{i_1, i_2, ...\} \in \widetilde{\mathcal{I}}_{ln}$. To be clear, the uncertain parameters fixed at node $(l,n)$ should have realized at the nodes in the layers $l' < l$ and must lie on the path connecting node $(l,n)$ and the root node. To summarize, the unresolved uncertain parameters are fixed to their expected values while the resolved uncertain parameters are fixed to their realized values. Since we utilize the expected value representation instead of scenarios, we no longer need the index $s$. For the same reason, we drop the NACs. Constraints (13b) and (13c) denote fixing of variables from the preceding time periods. The investment and operational decisions for a time period $t$ obtained at node $(l,n)$ are denoted by $x^*_{itln}$ and $y^*_{tln}$, respectively. It is important to note that at node $(l,n)$, we fix the investment decisions for time period $t$ such that $t \leq l-1$; however, we fix the operational decisions only for time period $t$ such that $t \leq l-2$. For example, in Figure 6, at node $(3,1)$, we fix $x_{i2} = x^*_{i221}, x_{i1} = x^*_{i121} \,\forall i$ and $y_1 = y^*_{121}$.

**Algorithm 1** Algorithm for Computing the VMSS with Endogenous Uncertainty

1: Solve (MSP), get $Z^{\text{MSP}}$
2: $l \leftarrow 1, \mathcal{N}_1 \leftarrow \{1\}$
3: **while** $l \leq T + 1$ **do**
4:     **for all** $n \in \mathcal{N}_l$ **do**
5:         Get $\widetilde{\mathcal{I}}_{ln}$ using (F-4)
6:         Get $p_{ln}$ using (F-5)
7:         $\theta_i \leftarrow \bar{\theta}_i \quad \forall i \in \mathcal{I} \backslash \widetilde{\mathcal{I}}_{ln}$               ▷ Assigning expected values
8:         $\theta_i \leftarrow \hat{\theta}_i^r \quad \forall i \in \widetilde{\mathcal{I}}_{ln}$                   ▷ Assigning realizations
9:         $x_{it} \leftarrow x_{it,l-1,\bar{n}_{ln}}^* \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, t \leq l-1$      ▷ Fixing investment decisions
10:        $y_t \leftarrow y_{t,l-1,\bar{n}_{ln}}^* \quad \forall t \in \mathcal{T}, t \leq l-2$          ▷ Fixing operational decisions
11:        Solve ($\text{DP}_{ln}$), get $Z_{ln}^{\text{DP}}, y_t^*$, and $x_{it}^* \quad \forall i \in \mathcal{I}, t \in \mathcal{T}$
12:        $x_{itln}^* \leftarrow x_{it}^* \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, t \leq l$        ▷ Storing investment decisions
13:        $y_{tln}^* \leftarrow y_t^* \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, t \leq l-1$      ▷ Storing operational decisions
14:        Get $\mathcal{I}'_{ln}$ using (F-0)
15:        Get $\sigma_{ln}$ using (F-1)
16:        Get $\mathcal{C}_{ln}$ using (F-2)
17:     **end for**
18:     Get $\mathcal{N}_{l+1}$ using (F-3)
19:     $l \leftarrow l + 1$
20: **end while**
21: $\overline{Z}^{\text{DP}} \leftarrow \sum_{n \in N_{T+1}} p_{T+1,n} Z_{T+1,n}^{\text{DP}}$
22: VMSS $= \overline{Z}^{\text{DP}} - Z^{\text{MSP}}$

The solution obtained from deterministic optimization is the probability-weighted average of the leaf nodes' solutions. Leaf nodes belong to the layer $l = T + 1$.

$$\overline{Z}^{\text{DP}} = \sum_{n \in \mathcal{N}_{T+1}} p_{T+1,n} Z_{T+1,n}^{\text{DP}}$$

Once we have the solutions to the stochastic problem and the deterministic problem, we can substitute them in equation (12) to obtain the value of stochastic solution for a multistage problem (VMSS) with endogenous uncertainty. For the detailed algorithm, see Algorithm 1.

$$\text{VMSS} = \overline{Z}^{\text{DP}} - Z^{\text{MSP}}$$

# 5 Column generation strategy

The stochastic programming models for large-scale problems are, in general, intractable due to a large number of scenarios and NACs. This demands the development of a decomposition strategy that can split the stochastic programming model into several smaller problems while at the same time preserving the optimal solution to the original model.

If we ignore the NACs (8a)-(8d), ($\text{FS}_{\text{SP}}$) decomposes into $|\mathcal{S}|$ independent subproblems. Notice that the constraint set (6) holds individually for each scenario $s \in \mathcal{S}$. For a scenario $s$, let $\mathcal{X}_s :=$

$\{x_s : (6)\}$ be the feasible space of binary variables $x_s \in \{0,1\}^{|\mathcal{T}| \sum_{k \in \mathcal{K}} |\mathcal{I}_k|}$. The binary restriction on variables $x$ ensures that the set $\mathcal{X}_s$ has finite cardinality $N_s := |\mathcal{X}_s|$; hence, we can rewrite $\mathcal{X}_s = \{x_{sc}^* : c \in \mathcal{C}_s\}$, where $\mathcal{C}_s = \{1, ..., N_s\}$. The selection of exactly one element (column) from set $\mathcal{X}_s$ can then be enforced as follows:

$$x_{kits} = \sum_{c \in \mathcal{C}_s} \lambda_{sc} x_{kitsc}^* \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k, t \in \mathcal{T} \tag{14}$$

$$\sum_{c \in \mathcal{C}_s} \lambda_{sc} = 1 \tag{15}$$

$$\lambda_{sc} \in \{0,1\} \quad \forall c \in \mathcal{C}_s \tag{16}$$

Further, each feasible column $x_{sc}^*$ is assumed to have an associated optimal cost $\Psi_{sc}^*$ obtained from solving the following optimization problem:

$$\Psi_{sc}^* = \min_{\substack{y,\Delta,C,P, \\ Q,V}} \; p_s \sum_{t \in \mathcal{T}} \alpha_t \left[ \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}_k} \left( \int_{\Phi_{k,i-1}}^{\Phi_{ki}} f_{ks}(\Phi_k) \mathrm{d}\Phi_k \right) x_{kits}^* + \right.$$
$$\left. n_t \sum_{h \in \mathcal{H}_t} \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} \left( \beta_{kjth} \rho_{kjt} P_{kths} + u_{kjth}(C_{kts}, Q_{jths}, P_{kths}, y_{kths}, ...) \right) \right]$$

$$
\left.
\begin{aligned}
\text{subject to} \quad & C_{k0s} = \overline{C}_{k0} \quad \forall k \in \mathcal{K} \\
& C_{kts} = C_{k,t-1,s} + \Delta_{kts} \quad \forall k \in \mathcal{K}, t \in \mathcal{T} \\
& \Delta_{kts} = \sum_{i \in \mathcal{I}_k} x_{kits}^* \overline{\Delta}_{ki} \quad \forall k \in \mathcal{K}, t \in \mathcal{T} \\
& \Delta_{kts} \leq b_{kt} \quad \forall k \in \mathcal{K}, t \in \mathcal{T} \\
& Q_{jths} = Q_{jt,h-1,s} + \sum_{k \in \mathcal{K}} \rho_{kjt} P_{kths} + V_{jths} - d_{jth} \quad \forall j \in \mathcal{J}, t \in \mathcal{T}, h \in \mathcal{H}_t \\
& Q_{jtH_ts} \geq Q_{jt0s} \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \\
& P_{kths} \leq \eta_{kth} C_{kts} \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, h \in \mathcal{H}_t \\
& g_{kths}(C_{kts}, Q_{jths}, P_{kths}, y_{kths}, ...) \leq 0 \quad \forall k \in \mathcal{K}, j \in \mathcal{J}, t \in \mathcal{T}, h \in \mathcal{H}_t \\
& 0 \leq V_{jths} \leq V_{jth}^{\max} \quad \forall j \in \mathcal{J}, t \in \mathcal{T}, h \in \mathcal{H}_t \\
& Q_{jths} \geq 0 \quad \forall j \in \mathcal{J}, t \in \mathcal{T}, h \in \mathcal{H}_t \\
& P_{kths} \geq 0, y_{kths} \in \mathbb{Z} \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, h \in \mathcal{H}_t \\
& C_{kts}, \Delta_{kts} \geq 0 \quad \forall k \in \mathcal{K}, t \in \mathcal{T}
\end{aligned}
\right\} \Xi
$$

Now that we have the set of equations (14)-(16) that enables us to select exactly one column from set $\mathcal{X}_s$ as well as the cost $\Psi_{sc}^*$ associated with it, (FS$_{\text{SP}}$) can thus be equivalently written as the following master problem (MP):

$$v^{\text{MP}} = \min_{\lambda,z} \; \sum_{s \in \mathcal{S}} \sum_{c \in \mathcal{C}_s} \lambda_{sc} \Psi_{sc}^* \tag{MP}$$

$$\text{subject to} \quad \sum_{c \in \mathcal{C}_s} \lambda_{sc} x_{ki1sc}^* - \sum_{c' \in \mathcal{C}_{s+1}} \lambda_{s+1,c'} x_{ki1,s+1,c'}^* = 0$$

$$\forall k \in \mathcal{K}, i \in \mathcal{I}_k, s \in \mathcal{S}, s < |\mathcal{S}| \qquad [\pi^{(1)}]$$

$$\sum_{c\in\mathcal{C}_s}\lambda_{sc}x^*_{ki,t+1,sc} - \sum_{c'\in\mathcal{C}_{s'}}\lambda_{s'c'}x^*_{ki,t+1,s'c'} \geq -(1-z_t^{s,s'})$$

$$\forall k\in\mathcal{K}, i\in\mathcal{I}_k, (s,s')\in\mathcal{P}', t\in\mathcal{T}\backslash\{T\} \qquad [\boldsymbol{\pi^{(2)}}]$$

$$\sum_{c\in\mathcal{C}_s}\lambda_{sc}x^*_{ki,t+1,sc} - \sum_{c'\in\mathcal{C}_{s'}}\lambda_{s'c'}x^*_{ki,t+1,s'c'} \leq (1-z_t^{s,s'})$$

$$\forall k\in\mathcal{K}, i\in\mathcal{I}_k, (s,s')\in\mathcal{P}', t\in\mathcal{T}\backslash\{T\} \qquad [\boldsymbol{\pi^{(3)}}]$$

$$z_t^{s,s'} + \sum_{c\in\mathcal{C}_s}\lambda_{sc}x^*_{ri\tau sc} \leq 1$$

$$\forall (s,s')\in\mathcal{P}', (r,i)\in\mathcal{D}(s,s'), \forall\tau, t\in\mathcal{T}\backslash\{T\}, \tau\leq t \qquad [\boldsymbol{\pi^{(4)}}]$$

$$z_t^{s,s'} + \sum_{(r,i)\in\mathcal{D}(s,s')}\sum_{\tau=1}^{t}\sum_{c\in\mathcal{C}_s}\lambda_{sc}x^*_{ri\tau sc} \geq 1 \quad \forall (s,s')\in\mathcal{P}', t\in\mathcal{T}\backslash\{T\} \qquad [\boldsymbol{\pi^{(5)}}]$$

$$\sum_{c\in\mathcal{C}_s}\lambda_{sc} = 1 \quad \forall s\in\mathcal{S} \qquad [\boldsymbol{\mu}]$$

$$\lambda_{sc}\in\{0,1\} \quad \forall s\in\mathcal{S}, c\in\mathcal{C}_s$$

$$z_t^{s,s'}\in\{0,1\} \quad \forall(s,s')\in\mathcal{P}', t\in\mathcal{T}\backslash\{T\}$$

where binary variable $z_t^{s,s'}$ corresponds to the Boolean variable $Z_t^{s,s'}$ in (FS$_{\text{SP}}$).

The above formulation is commonly known as a Dantzig-Wolfe reformulation, which often serves as the basis of a column generation algorithm (Barnhart et al., 1998; Lübbecke and Desrosiers, 2005). Now, the number of variables in (MP) can be extremely large due to the cardinality of $\mathcal{X}_s$; therefore, instead of solving (MP), we solve a restricted version of it, (RMP), with only a subset of columns present, i.e., $\overline{\mathcal{X}}_s \subseteq \mathcal{X}_s$. The columns to be included in (RMP) are determined by solving a set of pricing problems. The pricing problem for scenario $s$ is defined as follows:

$$\zeta_s = \min_{\substack{x,y,\Delta,C,P,\\Q,V}} \quad p_s\sum_{t\in\mathcal{T}}\alpha_t\left[\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{I}_k}\left(\int_{\Phi_{k,i-1}}^{\Phi_{ki}}f_{ks}(\Phi_k)\mathrm{d}\Phi_k\right)x_{kits}+ \right. \qquad \text{(PP}_\text{s}\text{)}$$

$$n_t\sum_{h\in\mathcal{H}_t}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}}\left(\beta_{kjth}\rho_{kjt}P_{kths} + u_{kjth}(C_{kts}, Q_{jths}, P_{kths}, y_{kths}, ...)\right)\bigg)\bigg]-$$

$$\sum_{\substack{k\in\mathcal{K},\\i\in\mathcal{I}_k}}\left[\left(\pi^{(1)}_{kis} - \pi^{(1)}_{ki,s-1}\right)x_{ki1s}\right]-$$

$$\sum_{\substack{k\in\mathcal{K},\\i\in\mathcal{I}_k,\\t\in\mathcal{T}\backslash\{T\}}}\left[\left(\sum_{(s,s')\in\mathcal{P}'}\pi^{(2)}_{kitss'} - \sum_{(s',s)\in\mathcal{P}'}\pi^{(2)}_{kits's}\right)x_{ki,t+1,s}\right]-$$

$$\sum_{\substack{k\in\mathcal{K},\\i\in\mathcal{I}_k,\\t\in\mathcal{T}\backslash\{T\}}}\left[\left(\sum_{(s,s')\in\mathcal{P}'}\pi^{(3)}_{kitss'} - \sum_{(s',s)\in\mathcal{P}'}\pi^{(3)}_{kits's}\right)x_{ki,t+1,s}\right]-$$

18

$$\sum_{\substack{\tau,t\in\mathcal{T}\backslash\{T\},\\ \tau\leq t}} \sum_{\substack{(s,s')\in\mathcal{P}',\\ (r,i)\in\mathcal{D}(s,s')}} \pi^{(4)}_{rit\tau ss'} x_{ri\tau s} - \sum_{\substack{t\in\mathcal{T}\backslash\{T\},\\ (s,s')\in\mathcal{P}'}} \pi^{(5)}_{tss'} \sum_{\substack{(r,i)\in\mathcal{D}(s,s'),\\ \tau\in\mathcal{T}\backslash\{T\},\tau\leq t}} x_{ri\tau s} - \mu_s$$

$$\text{subject to} \quad x_{kits} \leq \sum_{\tau=1}^{t} x_{k,i-1,\tau s} \quad \forall k\in\mathcal{K}, i\in\mathcal{I}_k\backslash\{1\}, t\in\mathcal{T}$$

$$\sum_{\tau=1}^{t} x_{ki\tau s} \leq 1 \quad \forall k\in\mathcal{K}, i\in\mathcal{I}_k, t\in\mathcal{T}$$

$$x_{kits}\in\{0,1\} \quad \forall k\in\mathcal{K}, i\in\mathcal{I}_k, t\in\mathcal{T}$$

$$(y,\Delta,C,P,Q,V)\in\Xi$$

where the dual prices $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$ correspond to the NACs and the convexity constraints in (MP), respectively. Further, if the reduced cost $\zeta_s$ is negative for some vector $x_s^*$, then it becomes eligible to enter the column set $\overline{\mathcal{X}}_s$ defining (RMP).

Once we have the master and subproblems defined, the algorithm requires solving the relaxation of (MP), which we denote (MP-LP). Each iteration of the column generation algorithm involves solving a linear relaxation of (RMP), which we denote (RMP-LP), with an updated column set $\overline{\mathcal{X}}_s$ from the pricing problems solved in the preceding iteration. The mutually independent pricing problems allow us to solve them in parallel, reducing computation time. Since (RMP-LP) contains only a subset of all columns, $v^{\text{RMP-LP}}$ is an upper bound ($\overline{\text{UB}}$) to (MP-LP). The lower bound ($\overline{\text{LB}}$) to (MP-LP) can be shown to be $\overline{\text{UB}} + \sum_{s\in\mathcal{S}}\zeta_s$ (Allman and Zhang, 2021).

The iterative column generation procedure continues until the optimality gap drops below the desired tolerance ($\bar{\epsilon}$). If column generation yields an integer feasible solution, then it is the optimal solution to (MP). However, in the case of fractional decisions, instead of branching, we simply solve (RMP) with the accumulated columns to quickly generate a feasible solution (UB) to (MP). Also, the lower bound to (MP-LP), $\overline{\text{LB}}$, is a lower bound to the (MP) as well, which allows us to compute a rigorous optimality gap for (MP). Figure 7 summarizes the column generation algorithm through a flow chart.

**Remark 1.** Column generation does not require us to solve the pricing problems to optimality. A sub-optimal column with a negative reduced cost is acceptable. Let $\overline{\text{LB}}_s$ be the lower bound associated with pricing problem $s$, then we get a valid lower bound on the (MP-LP) as follows:

$$v^{\text{RMP-LP}} + \sum_{s\in\mathcal{S}}\overline{\text{LB}}_s \leq v^{\text{MP-LP}} \leq v^{\text{RMP-LP}}$$

The above technique can speed up convergence, especially if the pricing problems are computationally difficult, by quickly generating columns that are feasible to the (RMP-LP). However, to prove convergence, we are eventually required to solve the pricing problems to optimality. The decision of when to start solving pricing problems to optimality is influenced by factors such as current gap and optimality tolerance, convergence rate, size of (RMP-LP), and iteration limit, to name a few.
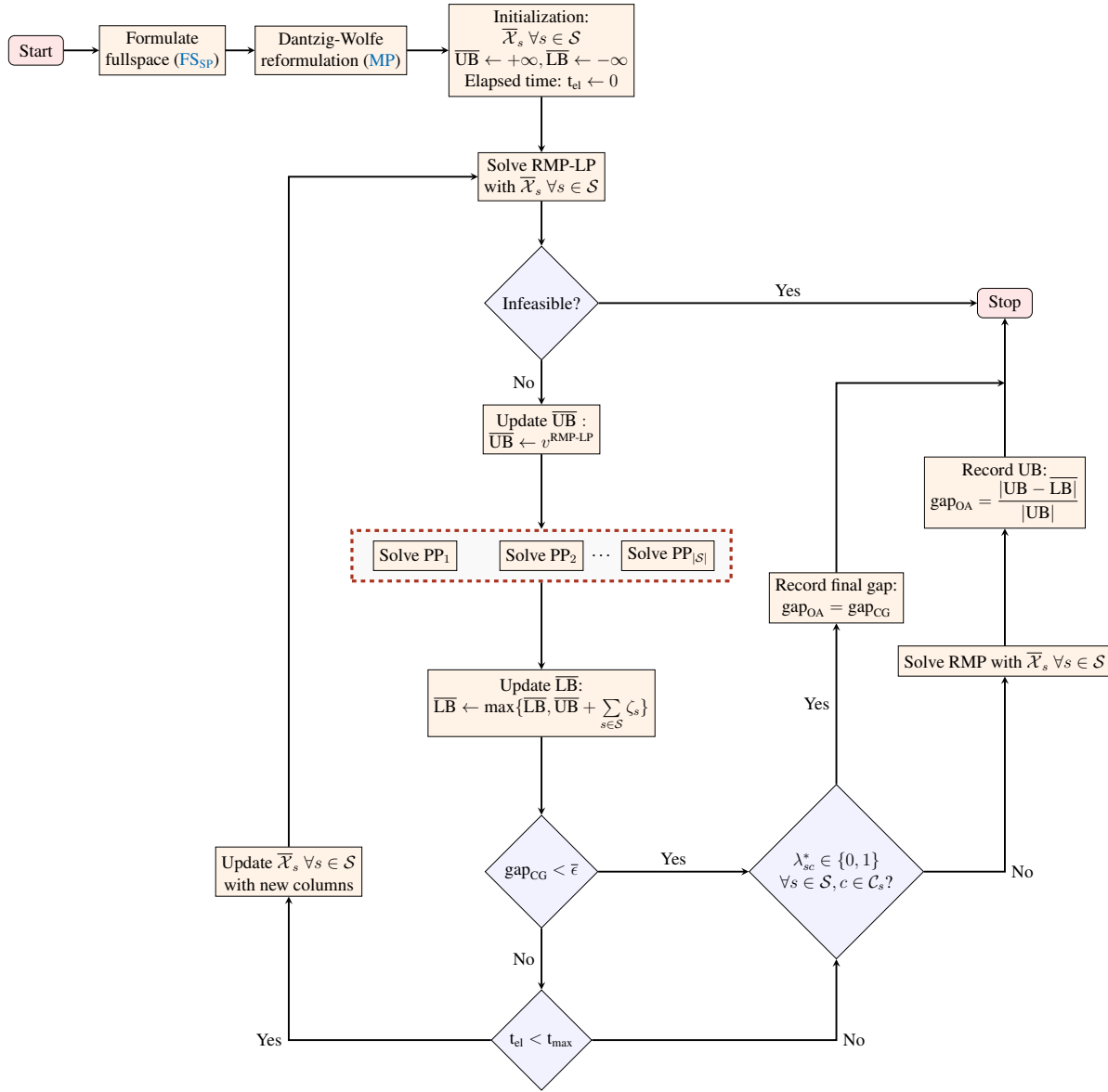
**Figure 7:** Flowchart depicting the steps involved in the proposed column generation algorithm. (gap$_{OA}$=overall feasible solution gap, gap$_{CG}$=relaxed solution gap at the end of column generation, $t_{el}$=elapsed time, $t_{max}$=maximum time limit. Although not explicitly shown, $t_{el}$ is updated at the end of each iteration.)

# 6 Computational experiments

This section evaluates the VMSS on randomly generated instances to showcase the benefits of modeling uncertainty using a stochastic programming framework. Further, an analysis of the performance difference between solving the full-space model and column generation is presented. All instances were modeled using JuMP v0.21.10 (Dunning et al., 2017) in Julia v1.6.3 (Bezanson et al., 2017). All instances were solved using Gurobi v9.1.2 (Gurobi Optimization, LLC, 2021) on the Mangi cluster of the Minnesota Supercomputing Institute (MSI) equipped with dual-socket AMD EPYC 7702 64-core processors. The number of cores utilized to solve each instance was set such that complete parallelization could be achieved. For all instances solved using the full-space model and column generation, the optimality tolerance was set to 0.1%. Further, a maximum time limit of 10,000 s was enforced for each instance. To counter the frequently observed tailing-off effect in column generation, we disabled presolve and crossover functionality, and solved (RMP-LP) using the barrier algorithm. The capacity planning model used in this section can be found in the supplementary material, Section C.

We consider instances with varying numbers of scenarios (8, 16, 32, and 64) and time periods (5, 7, and 9). The 8-, 16-, and 32-scenario instances were generated assuming a single uncertain technology, whereas the instances with 64 scenarios were generated assuming two uncertain technologies, each with 8 possible learning curves. For each case, the total numbers of technologies, $|\mathcal{K}|$, and products, $|\mathcal{J}|$, considered were four and one, respectively. The number of expansion points was assumed to be the same for all technologies, i.e., $|\mathcal{I}|_k = |\mathcal{I}|_{k'} \ \forall k, k' \in \mathcal{K}, k \neq k'$. In particular, instances with 8, 16, and 32 scenarios have $|\mathcal{I}|_k$ = 3, 4, and 5, respectively. For the 64-scenario case, for each technology, $|\mathcal{I}|_k$ equals 3. The frequency of representative day in each time period, $n_t$, was assumed to be 365, whereas the number of operational scheduling intervals, $|\mathcal{H}_t|$, was assumed to be 24 $\forall t \in \mathcal{T}$.

Demand data was generated separately for the initial and final time periods according to the distribution in Table 1. For the intermediate time periods, representative demand was distributed (and sorted in an increasing order) between the demand in the first and the last time period. Now, since each time period $t$ has $|\mathcal{H}_t|$ operational scheduling intervals, the demand in each $h \in \mathcal{H}_t$ was generated from $\mathcal{U}(0.8, 1.15)\bar{d}_t$, where $\bar{d}_t$ is the representative demand in time period $t$. To mimic the variable availability of renewable energy sources, we consider an availability parameter, $\eta_{kth}$, sampled from $\mathcal{U}(0.8, 1)$, denoting fraction of available installed capacity for technology $k$ in interval $h$ of time period $t$. The minimum up-time and down-time for each technology $k$ was generated according to the discrete distribution $\mathcal{U}\{1, 2\}$. The capacity associated with a single unit for each technology $k$ belongs to $\mathcal{U}(350, 450)$.

The learning curves for uncertain technologies were generated to represent both the high- and low-learning scenarios. For example, a high-learning scenario may correspond to a steep decline in cost initially (as much as 90-95%). Then with further increase in capacity, the rate of decline gradually reduces (0-60%). A low-learning scenario may correspond to virtually no reduction in capacity initially and then a slow decrease (0-30%) with any subsequent expansion. For deterministic technologies, the learning curves were generated such that they lie between the high and low learning scenarios of the uncertain technology. If the deterministic learning curves lie completely above or below the uncertain learning curves, that may entirely favor or disregard the expansion of uncertain technologies compared to the deterministic technologies, making the decision-making process trivial. This is a close representation of a real-world scenario, where a new technology

(potentially uncertain) with a cost initially higher than a matured technology (deterministic) may either undergo low-learning, in which case it fails to become competitive, or high-learning, in which case it eventually replaces the current technology.

| Parameter | Distribution |
|---|---|
| Initial capacity, $\overline{C}_{k0}$ | $\mathcal{U}(4,6)10^3$ |
| Incremental capacity, $\overline{\Delta}_{ki}$ | $\mathcal{U}(4,5)10^3$ |
| Initial expansion cost, $f_k(0)$ | $\mathcal{U}(4,6)10^6$ |
| Production cost, $\beta_{kth}$ | $\mathcal{U}(4,5)$ |
| Initial demand, $\bar{d}_1$ | $\mathcal{U}(0.8,1.3)(\sum_{k\in\mathcal{K}}\overline{C}_{k0})$ |
| Final demand, $\bar{d}_T$ | $\mathcal{U}(0.5,0.7)(\sum_{k\in\mathcal{K}}\overline{C}_{k|\mathcal{I}_k|})$ |
| Demand, $\bar{d}_t$ for $2 \leq t \leq T-1$ | $\mathcal{U}(\bar{d}_1,\bar{d}_T)$ |
| Budget, $b_{kt}$ | $\mathcal{U}(10,20)\mathbb{E}[\overline{\Delta}_{ki}]$ |
| Discount rate (%) | $\mathcal{U}(2,6)$ |
| Purchase/unmet demand cost, $\gamma_{th}$ | $\mathcal{U}(20,40)10^3$ |

Table 1: Distributions used for generating random problem instances.

## 6.1 Value of multistage stochastic solution

For each combination of $|\mathcal{S}|$ and $|\mathcal{T}|$, we solve 12 randomly generated instances. Table 2 summarizes the results for the case with one uncertain technology in terms of the average objective value from the expected value deterministic model (EV), the average objective value of the best solution we could find using stochastic programming (SP), and the mean, minimum, and maximum relative VMSS values.

| | | Mean obj. ($\times 10^6$) | | VMSS (%) | | |
|---|---|---|---|---|---|---|
| $|\mathcal{S}|$ | $|\mathcal{T}|$ | EV | SP | Mean | Min. | Max. |
| 8 | 5 | 73,278 | 70,386 | 3.91 | 1.10 | 6.83 |
| | 7 | 80,603 | 77,070 | 4.30 | 1.12 | 7.94 |
| | 9 | 82,277 | 78,973 | 4.02 | 1.17 | 7.15 |
| 16 | 5 | 96,229 | 89,983 | 6.40 | 0.88 | 11.13 |
| | 7 | 81,147 | 77,032 | 5.01 | 1.22 | 7.21 |
| | 9 | 92,558 | 88,254 | 4.62 | 3.20 | 8.22 |
| 32 | 5 | 98,425 | 92,382 | 6.25 | 2.56 | 9.18 |
| | 7 | 107,890 | 103,344 | 4.30 | 1.38 | 9.05 |
| | 9 | 104,050 | 99,996 | 4.06 | 0.47 | 7.24 |

Table 2: VMSS statistics for instances with one uncertain and three deterministic technologies. For each combination of $|\mathcal{S}|$ and $|\mathcal{T}|$, 12 random instances were solved.

Stochastic programming clearly outperforms the expected value deterministic approach in each

case leading to a positive VMSS. We observe an average VMSS in the range ~3-7%, which is quite significant, especially when considering that the cost in long-term capacity planning often amounts to millions or billions of dollars. For instances with two uncertain technologies (Table 3), we again see a considerable improvement in the solution indicated by the relative VMSS, further stressing the benefits reaped by modeling uncertainty using the stochastic programming approach. Moreover, on average, the relative VMSS appears to be higher in case of two uncertain technologies compared to only one, indicating a positive correlation between VMSS and the number of uncertain technologies. However, we should be cautious about generalizing this correlation because VMSS is affected by various factors including the scenario distribution and actual realizations as well as other deterministic model parameters.

| | Mean obj. ($\times 10^6$) | | VMSS (%) | | |
|---|---|---|---|---|---|
| $|\mathcal{T}|$ | EV | SP | Mean | Min | Max |
| 5 | 79,590 | 73,406 | 7.57 | 4.41 | 10.28 |
| 7 | 77,941 | 73,896 | 5.23 | 1.75 | 9.76 |
| 9 | 85,336 | 80,353 | 5.66 | 1.11 | 8.22 |

**Table 3:** VMSS statistics for instances with two uncertain and two deterministic technologies. All instances correspond to 64 scenarios resulting from the combination of 8 scenarios from each uncertain technology.

## 6.2 Performance analysis: full-space vs column generation

Considering the same set of random model instances, we now compare the computational performance between the full-space model and the proposed column generation algorithm. All computational statistics are shown in Tables 4 and 5.

The first key observation is that, for all cases, on average, the best feasible solution obtained using column generation is at least as good as the one obtained from the full-space model. The difference is especially prominent for larger instances with 32 and 64 scenarios. The average improvement observed for the 32- and 64-scenario instances range from 0.8-89% and 2.7-97%, respectively. The difference in feasible solution gaps between the full-space model ($\overline{\text{gap}}$) and column generation ($\overline{\text{gap}}_{\text{OA}}$) indicates the superiority of column generation in producing better feasible solutions. However, it should be noted that a lower gap does not necessarily imply a better feasible solution. For example, for the 8-scenario case, although $\overline{\text{gap}} < \overline{\text{gap}}_{\text{OA}}$, column generation produces the same or slightly better feasible solutions. It is due to this observation that we did not see the need to further implement a branching scheme in the spirit of a branch-and-price algorithm to further reduce the optimality gap.

Secondly, compared to the full-space model, column generation manages to solve significantly more instances. In particular, for the 8-, 12-, and 32-scenario cases, on average, the full-space model converged only for ~2 out of the 12 instances, whereas column generation converged for ~9 instances. For the 64-scenario case, the full-space model failed to solve any instances, whereas column generation converged for 7 out of 12 instances. Although convergence in the case of column generation does not necessarily guarantee a feasible solution, it does substantially reduce the overall computation time since the solution time for the final RMP is fairly short.

| | | Full-space | | | | Column generation | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\|\mathcal{S}\|$ | $\|\mathcal{T}\|$ | $\overline{\text{Obj.}}$ ($\times 10^6$) | NS | $\overline{\text{gap}}$ (%) | time (s) | $\overline{\text{Obj.}}$ ($\times 10^6$) | NS | $\overline{\text{gap}}_{\text{CG}}$ (%) | time (s) | $\overline{\text{gap}}_{\text{OA}}$ (%) | $\overline{\overline{\text{gap}}}_{\text{OA}}$(%) |
| 8 | 5 | 70,386 | 7 | 1.92 | 752 | 70,386 | 0 | - | 698 | 5.66 | 5.66 |
| | 7 | 77,099 | 7 | 3.71 | 2,716 | 77,070 | 0 | - | 2,233 | 4.54 | 4.54 |
| | 9 | 79,017 | 11 | 4.46 | 1,154 | 78,973 | 0 | - | 3,522 | 5.23 | 5.23 |
| 16 | 5 | 90,015 | 12 | 6.13 | - | 89,983 | 0 | - | 3,961 | 4.85 | 4.85 |
| | 7 | 77,203 | 12 | 7.95 | - | 77,032 | 0 | - | 4,704 | 5.99 | 5.99 |
| | 9 | 88,805 | 11 | 14.22 | 7,247 | 88,254 | 7 | 11.59 | 5,326 | 12.49 | 7.26 |
| 32 | 5 | 93,129 | 12 | 13.49 | - | 92,382 | 3 | 7.86 | 5,648 | 6.79 | 5.10 |
| | 7 | 117,678 | 12 | 26.33 | - | 103,344 | 9 | 13.78 | 6,230 | 16.20 | 9.02 |
| | 9 | 965,473 | 12 | 59.84 | - | 99,996 | 11 | 17.39 | 8,006 | 20.75 | 9.15 |

**Table 4:** Summary statistics highlighting the difference in computational performance between solving the full-space model and the column generation algorithm for one uncertain and three deterministic technologies. ($\overline{\text{Obj.}}$= average objective value over the 12 random instances, NS=number of instances not solved to 0.1% optimality gap in 10,000 s, time=average solution time for instances solved to 0.1% gap , $\overline{\text{gap}}$=average optimality gap for instances not solved to 0.1% gap in 10,000 s, $\overline{\text{gap}}_{\text{CG}}$=average optimality gap for column generation for instances not solved to 0.1% gap in 10,000 s, $\overline{\text{gap}}_{\text{OA}}$=average feasible solution gap for all 12 instances, $\overline{\overline{\text{gap}}}_{\text{OA}}$=average feasible solution gap for instances for which column generation converged.)

Lastly, for the instances that converged, column generation always terminates with a sub-10% gap, as indicated by the parameter $\overline{\overline{\text{gap}}}_{\text{OA}}$, which is a significant improvement over the feasible solutions obtained using the full-space model, again more so for the 32- and 64-scenario cases. Overall, column generation proves to be an efficient decomposition method for solving large instances of the proposed multistage stochastic programming model.

| | Full-space | | | | Column generation | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $\|\mathcal{T}\|$ | $\overline{\text{Obj.}}$ ($\times 10^6$) | NS | $\overline{\text{gap}}$(%) | time (s) | $\overline{\text{Obj.}}$ ($\times 10^6$) | NS | $\overline{\text{gap}}_{\text{CG}}$ (%) | time (s) | $\overline{\text{gap}}_{\text{OA}}$ (%) | $\overline{\overline{\text{gap}}}_{\text{OA}}$(%) |
| 5 | 75,450 | 12 | 13.06 | - | 73,406 | 0 | - | 3,882 | 6.86 | 6.86 |
| 7 | 1,015,983 | 12 | 59.40 | - | 73,896 | 5 | 7.90 | 5,731 | 11.60 | 9.35 |
| 9 | 3,124,639 | 12 | 86.93 | - | 80,353 | 9 | 12.19 | 5,204 | 15.92 | 9.56 |

**Table 5:** Summary statistics highlighting the difference in computational performance between solving the full-space model and the column generation algorithm for two uncertain and two deterministic technologies. All instances correspond to 64 scenarios resulting from the combination of 8 scenarios from each uncertain technology.

# 7 Industrial case study

The proposed stochastic programming framework is applied to a capacity expansion case study for a network of power generation technologies. Specifically, we consider seven technologies and categorize them into three categories – conventional (no cost reduction), deterministic (known learning curve), and uncertain technology (uncertain learning curve). Nuclear, coal, combined cycle gas turbine (CCGT), and open cycle gas turbine (OCGT) are considered conventional, onshore

wind and solar are assumed to be deterministic, and offshore wind is assumed to have an uncertain learning curve. The planning problem was modeled using JuMP v0.21.10 in Julia v1.6.3 and was solved using Gurobi v9.1.2. The model and data for this case study are partially adapted from Heuberger et al. (2017). The model can be found in the supplementary material, Section D.

The planning horizon spans eight 5-year time periods from 2015 to 2055. The capacity expansion decisions are made at the start of each of these time periods. Each time period comprises a representative scheduling horizon of 24 hours. Decisions made in each hour of the scheduling horizon include the amount of power generation, number of units to be started or shut down based on the minimum up- and down-time of each technology, inventory transfer based on the power generation and the demand satisfied, and so on. Figure 8 illustrates the eight possible learning curves for offshore wind technology and the eventual scenario tree based on the expansion decisions made. The scenario tree indicates that the offshore wind capacity increases by 2.5 GW in the first time period; however, as expected, we do not see any further expansion for the low-learning (high-cost scenarios) case. In contrast, for the high-learning (low-cost scenarios) case, the capacity further expands by 5.8 GW in the second time period, resulting in four scenario tree nodes. Besides, we obtained an ∼1.7% VMSS, which amounts to £2.16 billion savings over the deterministic approach.
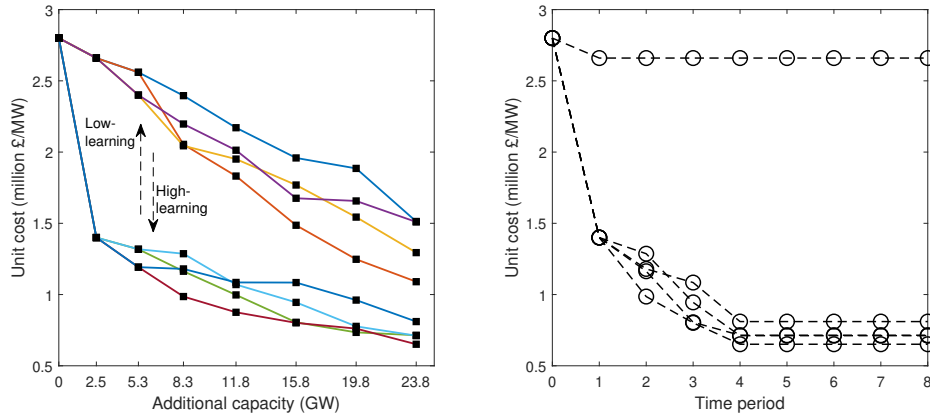


**Figure 8:** The possible learning curves considered for offshore wind are illustrated on the left. Scenario tree (right) obtained using the stochastic programming approach reveals expansion decisions for offshore wind.

Next, Figure 9 illustrates the distribution of capacity for all technologies during the planning horizon obtained using the deterministic and the stochastic programming approaches. Unlike stochastic programming, the deterministic approach fails to adapt its decisions to different learning rates and yields identical decisions under both the low- and high-learning scenarios, where no further investment is made in offshore wind . For the rest of this section, we will focus on discussing results obtained using the proposed stochastic programming framework. Now, compared to the high-learning scenario, the low-learning scenario does not favor offshore wind expansion. For the low-learning case, the mean installed offshore wind capacity during the planning period is 5 GW, which is 75% less than the high-learning case. This reduced capacity expansion in offshore wind is compensated by expansions of conventional technologies such as nuclear, CCGT, and OCGT. For example, in the low-learning case, nuclear power has a mean installed capacity of 13.8 GW, which is 34% higher than the high-learning case. Similarly, in the low-learning case, CCGT and

OCGT exceed the mean installed capacity in the high-learning case by 4% and 38%, respectively.

Further, since installed units of each technology have a finite lifetime, in the low-learning case, the minimal expansion of offshore wind results in retiring all its capacity by the end of 2040. Note that the expansion decisions are governed not only by the expansion cost but also by the expansion budget, production costs, lifetime of each technology, and the time-varying power generation capacity. The proposed stochastic programming model effectively integrates the above factors with the uncertain cost to generate the optimal capacity distribution.
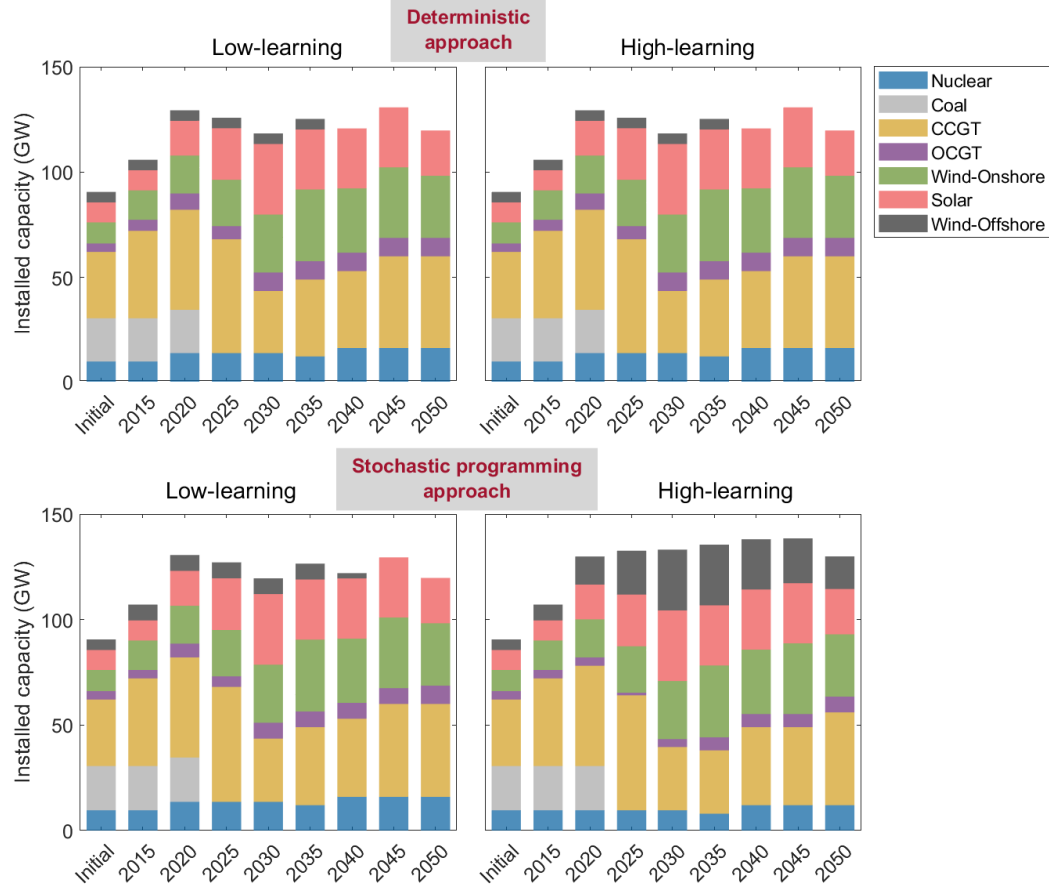


**Figure 9:** Capacity distribution of power generation technologies under low- and high-learning scenarios for the deterministic and stochastic programming approaches. Stochastic programming furnishes decisions adapted to the low- and high-learning scenarios, whereas deterministic approach fails to adapt and generates identical decisions for the two cases.

Now, the available capacity of a technology along with the production costs directly affect the amount of energy production that can be achieved in any particular hour of the day. Figure 10 demonstrates the effect of uncertainty in technology learning on the power generation through offshore wind technology in two distant time periods. In the case of low-learning, lower installed offshore wind capacity leads to lower power generation. In particular, the mean energy production during the day in 2020 is 1,475 MWh. Since most offshore wind capacity retires by 2040, the mean production reduces by 67% to 492 MWh. On the contrary, in the high-learning case, increased capacity expansion enables higher power generation capability to meet the increased demand. For

example, the mean power generation in 2020 is 2,616 MWh, which further increases by 79% to 4,681 MWh in 2040.
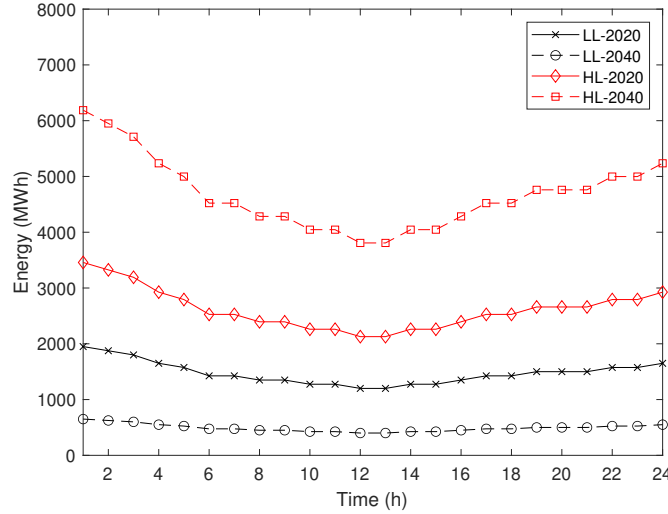


**Figure 10:** Offshore wind energy distribution obtained using the stochastic programming approach during a representative day in 2020 and 2040 under high-learning (HL) and low-learning (LL) cases.

# 8   Conclusions

In this work, we proposed a rigorous optimization framework for a general process network that can be utilized to model energy systems containing both renewable and non-renewable technologies. We applied stochastic programming to account for the long-neglected aspect of uncertainty in technology learning curves. Moreover, we developed an algorithm to compute the value of stochastic solution in multistage stochastic programming with type-2 endogenous uncertainty. We further developed a decomposition algorithm based on column generation to solve large instances. Improvement in tractability, especially for instances with a large number of scenarios, and solution quality relative to the full-space model was shown by applying our decomposition algorithm to a large set of randomly generated instances.

The practical applicability of the proposed framework was established through a case study on power capacity expansion. The difference in decisions was discussed primarily through two sub-classes of scenarios, high- and low-learning, also indicating that any solution obtained through a deterministic model, which essentially operates on the expected value of unrealized uncertain parameters, would often be sub-optimal for any perturbation in the assumed deterministic learning curves. Overall, the results demonstrate the importance of accounting for endogenous uncertainty in technology learning.

# Acknowledgments

# References

Ahmed, S., 2000. *Strategic planning under uncertainty: Stochastic integer programming approaches*. PhD thesis. University of Illinois at Urbana-Champaign.

Aliabadi, D. E., 2020. Decarbonizing existing coal-fired power stations considering endogenous technology learning: A Turkish case study. *Journal of Cleaner Production*, 261:121100.

Allman, A. and Zhang, Q., 2021. Branch-and-price for a class of nonconvex mixed-integer nonlinear programs. *Journal of Global Optimization*, 81(4):861–880.

Anzanello, M. J. and Fogliatto, F. S., 2011. Learning curve models and applications: Literature review and research directions. *International Journal of Industrial Ergonomics*, 41(5):573–583.

Apap, R. M. and Grossmann, I. E., 2017. Models and computational strategies for multistage stochastic programming under endogenous and exogenous uncertainties. *Computers & Chemical Engineering*, 103:233–274.

Bakker, S. J., Wang, A., and Gounaris, C. E., 2021. Vehicle routing with endogenous learning: Application to offshore plug and abandonment campaign planning. *European Journal of Operational Research*, 289(1):93–106.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W., and Vance, P. H., 1998. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329.

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B., 2017. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98.

Birge, J. R. and Louveaux, F., 2011. *Introduction to stochastic programming*. Springer Science & Business Media.

Chen, Y., Zhang, Y., Fan, Y., Hu, K., and Zhao, J., 2017. A dynamic programming approach for modeling low-carbon fuel technology adoption considering learning-by-doing effect. *Applied Energy*, 185:825–835.

Colvin, M. and Maravelias, C. T., 2008. A stochastic programming approach for clinical trial planning in new drug development. *Computers & Chemical Engineering*, 32(11):2626–2642.

Colvin, M. and Maravelias, C. T., 2010. Modeling methods and a branch and cut algorithm for pharmaceutical clinical trial planning using stochastic programming. *European Journal of Operational Research*, 203(1):205–215.

Daugaard, T., Mutti, L. A., Wright, M. M., Brown, R. C., and Componation, P., 2015. Learning rates and their impacts on the optimal capacities and production costs of biorefineries. *Biofuels, Bioproducts and Biorefining*, 9(1):82–94.

Dunning, I., Huchette, J., and Lubin, M., 2017. JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320.

Escudero, L. F., Garín, A., Merino, M., and Pérez, G., 2007. The value of the stochastic solution in multistage problems. *Top*, 15(1):48–64.

Goel, V. and Grossmann, I. E., 2004. A stochastic programming approach to planning of offshore gas field developments under uncertainty in reserves. *Computers & Chemical Engineering*, 28 (8):1409–1429.

Goel, V. and Grossmann, I. E., 2006. A class of stochastic programs with decision dependent uncertainty. *Mathematical Programming*, 108(2):355–394.

Gupta, V. and Grossmann, I. E., 2014. Multistage stochastic programming approach for offshore oilfield infrastructure planning under production sharing agreements and endogenous uncertainties. *Journal of Petroleum Science and Engineering*, 124:180–197.

Gurobi Optimization, LLC, 2021. Gurobi Optimizer Reference Manual.

Hellemo, L., Barton, P. I., and Tomasgard, A., 2018. Decision-dependent probabilities in stochastic programs with recourse. *Computational Management Science*, 15(3):369–395.

Heuberger, C. F., Rubin, E. S., Staffell, I., Shah, N., and Mac Dowell, N., 2017. Power capacity expansion planning considering endogenous technology cost learning. *Applied Energy*, 204: 831–845.

Hooshmand, F. and MirHassani, S., 2016. Efficient constraint reduction in multistage stochastic programming problems with endogenous uncertainty. *Optimization Methods and Software*, 31 (2):359–376.

Jonsbråten, T. W., Wets, R. J., and Woodruff, D. L., 1998. A class of stochastic programs with decision dependent random elements. *Annals of Operations Research*, 82:83–106.

Kim, S., Koo, J., Lee, C. J., and Yoon, E. S., 2012. Optimization of Korean energy planning for sustainability considering uncertainties in learning rates and external factors. *Energy*, 44(1): 126–134.

Laffel, G. L., Barnett, A. I., Finkelstein, S., and Kaye, M. P., 1992. The relation between experience and outcome in heart transplantation. *New England Journal of Medicine*, 327(17):1220–1225.

Lieberman, M. B., 1984. The learning curve and pricing in the chemical processing industries. *The RAND Journal of Economics*, 15(2):213–228.

Lübbecke, M. E. and Desrosiers, J., 2005. Selected topics in column generation. *Operations Research*, 53(6):1007–1023.

Maggioni, F., Allevi, E., and Bertocchi, M., 2014. Bounds in multistage linear stochastic programming. *Journal of Optimization Theory and Applications*, 163(1):200–229.

Nordhaus, W. D., 2014. The perils of the learning model for modeling endogenous technological change. *The Energy Journal*, 35(1):1–13.

Peeta, S., Salman, F. S., Gunnec, D., and Viswanath, K., 2010. Pre-disaster investment decisions for strengthening a highway network. *Computers & Operations Research*, 37(10):1708–1719.

Rout, U. K., Blesl, M., Fahl, U., Remme, U., and Voß, A., 2009. Uncertainty in the learning rates of energy technologies: An experiment in a global multi-regional energy system model. *Energy Policy*, 37(11):4927–4942.

Rubin, E. S., 2019. Improving cost estimates for advanced low-carbon power plants. *International Journal of Greenhouse Gas Control*, 88:1–9.

Rubin, E. S., Yeh, S., Antes, M., Berkenpas, M., and Davison, J., 2007. Use of experience curves to estimate the future cost of power plants with $CO_2$ capture. *International Journal of Greenhouse Gas Control*, 1(2):188–197.

Strunge, T., Renforth, P., and Van der Spek, M., 2022. Towards a business case for $CO_2$ mineralisation in the cement industry. *Communications Earth & Environment*, 3(1):1–14.

Sturm, R., 1999. Cost and quality trends under managed care: is there a learning curve in behavioral health carve-out plans? *Journal of Health Economics*, 18(5):593–604.

Van der Zwaan, B. and Rabl, A., 2003. Prospects for PV: a learning curve analysis. *Solar energy*, 74(1):19–31.

Wright, T. P., 1936. Factors affecting the cost of airplanes. *Journal of the Aeronautical Sciences*, 3(4):122–128.

Zhang, Q. and Feng, W., 2020. A unified framework for adjustable robust optimization with endogenous uncertainty. *AIChE Journal*, 66(12):e17047.

Zhang, Q., Bremen, A. M., Grossmann, I. E., and Pinto, J. M., 2018. Long-term electricity procurement for large industrial consumers under uncertainty. *Industrial & Engineering Chemistry Research*, 57(9):3333–3347.

# Supplementary material

# Capacity planning with uncertain endogenous technology learning

Tushar Rathi and Qi Zhang

*Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, MN 55455, USA*

## A. Reformulation of disjunctive constraints (8b)

In this section, we reformulate the disjunctive constraints (8b) into linear constraints using the big-M reformulation technique.

$$
\begin{bmatrix} Z_t^{s,s'} \\ x_{ki,t+1,s} = x_{ki,t+1,s'} \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k \end{bmatrix} \vee \begin{bmatrix} \neg Z_t^{s,s'} \end{bmatrix} \forall (s,s') \in \mathcal{P}', t \in \mathcal{T} \backslash \{T\} \tag{8b}
$$

First, let binary variable $z_t^{s,s'}$ correspond to Boolean variable $Z_t^{s,s'}$ and binary variable $\bar{z}_t^{s,s'}$ correspond to Boolean variable $\neg Z_t^{s,s'}$. Now, in any feasible solution, only one of the two alternatives in the disjunction can be true; therefore, both binary variables must sum to 1:

$$
z_t^{s,s'} + \bar{z}_t^{s,s'} = 1 \quad \forall (s,s') \in \mathcal{P}', t \in \mathcal{T} \backslash \{T\} \tag{A2}
$$

Next, rewriting the equality constraints inside the first disjunct of disjunction (8b) as inequalities:

$$
x_{ki,t+1,s} = x_{ki,t+1,s'} \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k
$$
$$
\Rightarrow \quad x_{ki,t+1,s} - x_{ki,t+1,s'} \geq 0 \quad \text{and} \quad x_{ki,t+1,s} - x_{ki,t+1,s'} \leq 0 \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k \tag{A3}
$$

Now, let $M$ be a sufficiently large constant, then the disjunctive constraints (8b) can be written as the following linear constraints:

$$
x_{ki,t+1,s} - x_{ki,t+1,s'} \geq -M(1 - z_t^{s,s'}) \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k, (s,s') \in \mathcal{P}', t \in \mathcal{T} \backslash \{T\}
$$
$$
x_{ki,t+1,s} - x_{ki,t+1,s'} \leq M(1 - z_t^{s,s'}) \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k, (s,s') \in \mathcal{P}', t \in \mathcal{T} \backslash \{T\} \tag{A4}
$$

It is easy to verify that constraints (A4) collapse to constraints (A3) when $z_t^{s,s'}$ equals 1. Now, $x_{ki,t+1,s} - x_{ki,t+1,s'} \in \{-1, 0, 1\}$ because $x$ are binary variables; therefore, we can set $M$ to 1. Additionally, if $Z_t^{s,s'}$ is `false`, then no constraints apply. Therefore, we can ignore constraints (A2) (the binary restriction on variables $z$ is sufficient to satisfy these constraints). Finally, the linear constraints equivalent to disjunctive constraints (8b) are as follows:

$$x_{ki,t+1,s} - x_{ki,t+1,s'} \geq -(1 - z_t^{s,s'}) \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k, (s,s') \in \mathcal{P}', t \in \mathcal{T}\backslash\{T\}$$

$$x_{ki,t+1,s} - x_{ki,t+1,s'} \leq (1 - z_t^{s,s'}) \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k, (s,s') \in \mathcal{P}', t \in \mathcal{T}\backslash\{T\} \qquad \text{(A5)}$$

$$z_t^{s,s'} \in \{0,1\} \quad \forall (s,s') \in \mathcal{P}', t \in \mathcal{T}\backslash\{T\}$$

## B. Reformulation of propositional logic constraints (8c)

In this section, we reformulate logic constraints (8c) into linear constraints.

$$Z_t^{s,s'} \iff \bigwedge_{(r,i)\in\mathcal{D}(s,s')} \left[\bigwedge_{\tau=1}^{t}(\neg x_{ri\tau s})\right] \forall (s,s') \in \mathcal{P}', t \in \mathcal{T}\backslash\{T\} \qquad \text{(8c)}$$

Let binary variable $z_t^{s,s'}$ correspond to Boolean variable $Z_t^{s,s'}$.

**Forward implication:**

$$z_t^{s,s'} \Rightarrow \bigwedge_{(r,i)\in\mathcal{D}(s,s')} \left[\bigwedge_{\tau=1}^{t}(\neg x_{ri\tau s})\right] \qquad\qquad \forall (s,s') \in \mathcal{P}', t \in \mathcal{T}\backslash\{T\}$$

$$\Rightarrow \neg z_t^{s,s'} \vee \bigwedge_{(r,i)\in\mathcal{D}(s,s')} \left[\bigwedge_{\tau=1}^{t}(\neg x_{ri\tau s})\right] \qquad\qquad \forall (s,s') \in \mathcal{P}', t \in \mathcal{T}\backslash\{T\}$$

$$\Rightarrow \bigwedge_{(r,i)\in\mathcal{D}(s,s')} \bigwedge_{\tau=1}^{t} (\neg z_t^{s,s'} \vee \neg x_{ri\tau s}) \qquad\qquad \forall (s,s') \in \mathcal{P}', t \in \mathcal{T}\backslash\{T\}$$

$$\Rightarrow 1 - z_t^{s,s'} + 1 - x_{ri\tau s} \geq 1 \qquad\qquad \forall (s,s') \in \mathcal{P}', (r,i) \in \mathcal{D}(s,s'), \forall \tau, t \in \mathcal{T}\backslash\{T\}, \tau \leq t$$

$$\therefore z_t^{s,s'} \leq 1 - x_{ri\tau s} \qquad\qquad \forall (s,s') \in \mathcal{P}', (r,i) \in \mathcal{D}(s,s'), \forall \tau, t \in \mathcal{T}\backslash\{T\}, \tau \leq t$$

**Backward implication:**

$$\bigwedge_{(r,i)\in\mathcal{D}(s,s')} \left[\bigwedge_{\tau=1}^{t}(\neg x_{ri\tau s})\right] \Rightarrow z_t^{s,s'} \qquad\qquad \forall (s,s') \in \mathcal{P}', t \in \mathcal{T}\backslash\{T\}$$

$$\Rightarrow \neg\left[\bigwedge_{(r,i)\in\mathcal{D}(s,s')} \left[\bigwedge_{\tau=1}^{t}(\neg x_{ri\tau s})\right]\right] \vee z_t^{s,s'} \qquad\qquad \forall (s,s') \in \mathcal{P}', t \in \mathcal{T}\backslash\{T\}$$

$$\Rightarrow \bigvee_{(r,i)\in\mathcal{D}(s,s')} \left[\bigvee_{\tau=1}^{t}\neg(\neg x_{ri\tau s})\right] \vee z_t^{s,s'} \qquad\qquad \forall (s,s') \in \mathcal{P}', t \in \mathcal{T}\backslash\{T\}$$

$$\Rightarrow \bigvee_{(r,i)\in\mathcal{D}(s,s')} \left[\bigvee_{\tau=1}^{t}(x_{ri\tau s})\right] \vee z_t^{s,s'} \qquad\qquad \forall (s,s') \in \mathcal{P}', t \in \mathcal{T}\backslash\{T\}$$

$$\Rightarrow \sum_{(r,i)\in\mathcal{D}(s,s')} \sum_{\tau=1}^{t} x_{ri\tau s} + z_t^{s,s'} \geq 1 \qquad\qquad \forall (s,s') \in \mathcal{P}', t \in \mathcal{T}\backslash\{T\}$$

2

$$\therefore z_t^{s,s'} \geq 1 - \sum_{(r,i)\in\mathcal{D}(s,s')} \sum_{\tau=1}^{t} x_{ri\tau s} \qquad \forall (s,s') \in \mathcal{P}', t \in \mathcal{T}\backslash\{T\}$$

Thus, logic constraints (8c) can be replaced by linear constraints (B1) and (B2). Furthermore, $z_t^{s,s'}$ can be relaxed to interval [0,1] because it is forced to take binary values due to the binary restriction on variable $x_{rits}$ as well as constraints (B1) and (B2).

$$z_t^{s,s'} \leq 1 - x_{ri\tau s} \qquad \forall (s,s') \in \mathcal{P}', (r,i) \in \mathcal{D}(s,s'), \forall \tau, t \in \mathcal{T}\backslash\{T\}, \tau \leq t \quad \text{(B1)}$$

$$z_t^{s,s'} \geq 1 - \sum_{(r,i)\in\mathcal{D}(s,s')} \sum_{\tau=1}^{t} x_{ri\tau s} \qquad \forall (s,s') \in \mathcal{P}', t \in \mathcal{T}\backslash\{T\} \quad \text{(B2)}$$

## C. Model for computational experiments

In this section, we present the stochastic programming model developed for use in Section 6. Since we only considered a single product/resource, we ignore the index $j$ henceforth in our formulation. To model the operational part of the problem, we introduce the following additional parameters and variables. The capacity per unit of technology $k$ is represented by $\Omega_k$. The cost of unmet demand (or purchasing the product) in scheduling interval $h$ of time period $t$ is denoted by $\gamma_{kh}$. The minimum up- and down-time of technology $k$ are denoted by $UT_k$ and $DT_k$, respectively. The number of units of technology $k$ operating, starting up, and shut down in scheduling interval $h$ of time period $t$ and scenarios $s$ are represented by variables $y_{kths}$, $w_{kths}$, and $w'_{kths}$, respectively. The model is then written as follows:

$$\text{minimize} \quad \sum_{s\in\mathcal{S}} p_s \Bigg[ \sum_{t\in\mathcal{T}} \alpha_t \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}_k} \bigg( \int_{\Phi_{k,i-1}}^{\Phi_{ki}} f_{ks}(\Phi_k)\mathrm{d}\Phi_k \bigg) x_{kits} +$$

$$\sum_{t\in\mathcal{T}} \alpha_t n_t \sum_{h\in\mathcal{H}_t} \sum_{k\in\mathcal{K}} \beta_{kth} P_{kths} + \sum_{t\in\mathcal{T}} n_t \sum_{h\in\mathcal{H}_t} \gamma_{th} V_{ths} \Bigg] \qquad \text{(C0)}$$

subject to
$$C_{k0s} = \overline{C}_{k0} \quad \forall k \in \mathcal{K}, s \in \mathcal{S} \qquad \text{(C1)}$$

$$C_{kts} = C_{k,t-1,s} + \Delta_{kts} \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S} \qquad \text{(C2)}$$

$$\Delta_{kts} = \sum_{i\in\mathcal{I}_k} x_{kits}\overline{\Delta}_{ki} \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S} \qquad \text{(C3)}$$

$$\Delta_{kts} \leq b_{kt} \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S} \qquad \text{(C4)}$$

$$x_{kits} \leq \sum_{\tau=1}^{t} x_{k,i-1,\tau s} \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k\backslash\{1\}, t \in \mathcal{T}, s \in \mathcal{S} \qquad \text{(C5)}$$

$$\sum_{\tau=1}^{t} x_{ki\tau s} \leq 1 \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k, t \in \mathcal{T}, s \in \mathcal{S} \qquad \text{(C6)}$$

$$Q_{ths} = Q_{t,h-1,s} + \sum_{k\in\mathcal{K}} P_{kths} + V_{ths} - d_{th} \quad \forall t \in \mathcal{T}, h \in \mathcal{H}_t, s \in \mathcal{S} \qquad \text{(C7)}$$

$$Q_{tHts} \geq Q_{t0s} \quad t \in \mathcal{T}, s \in \mathcal{S} \qquad \text{(C8)}$$

$$\Omega_k y_{kths} \leq C_{kts} \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, h \in \mathcal{H}_t, s \in \mathcal{S} \qquad \text{(C9)}$$

$$w_{kths} = y_{kths} - y_{kt,h-1,s} \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, h \in \mathcal{H}_t, s \in \mathcal{S} \tag{C10}$$

$$w'_{kths} = y_{kt,h-1,s} - y_{kths} \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, h \in \mathcal{H}_t, s \in \mathcal{S} \tag{C11}$$

$$w_{kths} \leq y_{kth's}$$
$$\forall k \in \mathcal{K}, t \in \mathcal{T}, \ h, h' \in \mathcal{H}_t, h' \in \{h, h+1, ..., h + UT_k - 1\}, s \in \mathcal{S} \tag{C12}$$

$$w'_{kths} \leq \frac{C_{kts}}{\Omega_k} - y_{kth's}$$
$$\forall k \in \mathcal{K}, t \in \mathcal{T}, \ h, h' \in \mathcal{H}_t, h' \in \{h, h+1, ..., h + DT_k - 1\}, s \in \mathcal{S} \tag{C13}$$

$$P_{kths} \leq \eta_{kth} \Omega_k y_{kths} \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, h \in \mathcal{H}_t, s \in \mathcal{S} \tag{C14}$$

$$Q_{ths} \geq 0, V_{ths} \geq 0 \quad \forall t \in \mathcal{T}, h \in \mathcal{H}_t, s \in \mathcal{S} \tag{C15}$$

$$P_{kths} \geq 0, w_{kths} \in \mathbb{R}, w'_{kths} \in \mathbb{R}, y_{kths} \in \mathbb{Z} \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, h \in \mathcal{H}_t, s \in \mathcal{S} \tag{C16}$$

$$C_{kts}, \Delta_{kts} \geq 0 \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S} \tag{C17}$$

$$x_{kits} \in \{0, 1\} \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k, t \in \mathcal{T}, s \in \mathcal{S} \tag{C18}$$

NACs (8a) - (8d)

where the objective function (C0) minimizes the expected net expenditure. Constraints (C1)-(C3) and (C4)-(C6) represent the capacity balance and budget/expansion restrictions, respectively. The inventory balance is modeled in constraints (C7) and (C8). Constraints (C9)-(C13) help model the number of units of technology $k$ that operate, start or shut down in each scheduling interval of time period $t$ and scenario $s$ utilizing the information including capacity per unit and up-time/down-time of technology $k$. The time-varying availability of the installed capacity is enforced through constraints (C14). Constraints (C15)-(C18) enforce the bounds on model variables. Note that the above formulation is a mixed-integer linear programming (MILP) model owing to the binary and integer restrictions on variables $x$ and $y$, respectively.

## D. Model for industrial case study

In this section, we present the capacity planning model used in Section 7. Note that the model presented here is an extension of the one used for the computational experiments (see the previous section). The lifetime of the initial installed capacity of technology $k$ is represented as $LT_k^0$, whereas $LT_k$ represents the lifetime of capacity installed during the planning period. The length of each time period is denoted by $\delta_t$. The net capacity margin, maximum capacity provision in technology $k$, and the peak load in time period $t$ are denoted by $\overline{M}$, $\Gamma_k$, and $L_t^{\max}$, respectively. Lastly, $\Lambda_k^{\min}$ and $\Lambda_k^{\max}$ represent the minimum and maximum power output of technology $k$, respectively. The power capacity planning model is then defined as follows:

$$\text{minimize} \quad \sum_{s \in \mathcal{S}} p_s \Bigg[ \sum_{t \in \mathcal{T}} \alpha_t \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}_k} \bigg( \int_{\Phi_{k,i-1}}^{\Phi_{ki}} f_{ks}(\Phi_k) \mathrm{d}\Phi_k \bigg) x_{kits} +$$

$$\sum_{t \in \mathcal{T}} \alpha_t n_t \sum_{h \in \mathcal{H}_t} \sum_{k \in \mathcal{K}} \beta_{kth} P_{kths} + \sum_{t \in \mathcal{T}} n_t \sum_{h \in \mathcal{H}_t} \gamma_{th} V_{ths} \Bigg] \tag{D0}$$

$$\text{subject to} \quad C_{k0s} = \overline{C}_{k0} \quad \forall k \in \mathcal{K}, s \in \mathcal{S} \tag{D1}$$

$$C_{kts} = C_{k,t-1,s} + \Delta_{kts}$$

$$\forall k \in \mathcal{K}, t \in \{t'|t' \in \mathcal{T} \wedge t' \leq \frac{LT_k^0}{\delta_t} \vee \frac{LT_k^0}{\delta_t} + 1 < t' \leq \frac{LT_k}{\delta_t}\}, s \in \mathcal{S} \qquad \text{(D2)}$$

$$C_{kts} = C_{k,t-1,s} + \Delta_{kts} - \overline{C}_{k0} \quad \forall k \in \mathcal{K}, t = \frac{LT_k^0}{\delta_t} + 1, s \in \mathcal{S} \qquad \text{(D3)}$$

$$C_{kts} = C_{k,t-1,s} + \Delta_{kts} - \Delta_{k,t-\tau,s}$$
$$\forall k \in \mathcal{K}, t \in \{t'|t' \in \mathcal{T} \wedge t' > \frac{LT_k}{\delta_t}\}, \tau = \frac{LT_k}{\delta_t}, s \in \mathcal{S} \qquad \text{(D4)}$$

$$\Delta_{kts} = \sum_{i \in \mathcal{I}_k} x_{kits} \overline{\Delta}_{ki} \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S} \qquad \text{(D5)}$$

$$x_{kits} \leq \sum_{\tau=1}^{t} x_{k,i-1,\tau s} \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k \backslash \{1\}, t \in \mathcal{T}, s \in \mathcal{S} \qquad \text{(D6)}$$

$$\sum_{\tau=1}^{t} x_{ki\tau s} \leq 1 \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k, t \in \mathcal{T}, s \in \mathcal{S} \qquad \text{(D7)}$$

$$\Delta_{kts} \leq b_{kt}\delta_t \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S} \qquad \text{(D8)}$$

$$Q_{ths} = Q_{t,h-1,s} + \sum_{k \in \mathcal{K}} P_{kths} + V_{ths} - d_{th} \quad \forall t \in \mathcal{T}, h \in \mathcal{H}_t, s \in \mathcal{S} \qquad \text{(D9)}$$

$$Q_{tH_ts} \geq Q_{t0s} \quad t \in \mathcal{T}, s \in \mathcal{S} \qquad \text{(D10)}$$

$$\sum_{k \in \mathcal{K}} C_{kts}\Gamma_k \geq L_t^{\max}(1 + \overline{M}) \quad \forall t \in \mathcal{T}, s \in \mathcal{S} \qquad \text{(D11)}$$

$$\Omega_k y_{kths} \leq C_{kts} \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, h \in \mathcal{H}_t, s \in \mathcal{S} \qquad \text{(D12)}$$

$$w_{kths} = y_{kths} - y_{kt,h-1,s} \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, h \in \mathcal{H}_t, s \in \mathcal{S} \qquad \text{(D13)}$$

$$w'_{kths} = y_{kt,h-1,s} - y_{kths} \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, h \in \mathcal{H}_t, s \in \mathcal{S} \qquad \text{(D14)}$$

$$w_{kths} \leq y_{kth's}$$
$$\forall k \in \mathcal{K}, t \in \mathcal{T}, \ h, h' \in \mathcal{H}_t, h' \in \{h, h+1, ..., h+UT_k-1\}, s \in \mathcal{S} \qquad \text{(D15)}$$

$$w'_{kths} \leq \frac{C_{kts}}{\Omega_k} - y_{kth's}$$
$$\forall k \in \mathcal{K}, t \in \mathcal{T}, \ h, h' \in \mathcal{H}_t, h' \in \{h, h+1, ..., h+DT_k-1\}, s \in \mathcal{S} \qquad \text{(D16)}$$

$$P_{kths} \leq \eta_{kth}\Omega_k\Lambda_k^{\max}y_{kths} \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, h \in \mathcal{H}_t, s \in \mathcal{S} \qquad \text{(D17)}$$

$$P_{kths} \geq \eta_{kth}\Omega_k\Lambda_k^{\min}y_{kths} \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, h \in \mathcal{H}_t, s \in \mathcal{S} \qquad \text{(D18)}$$

$$Q_{ths} \geq 0, V_{ths} \geq 0 \quad \forall t \in \mathcal{T}, h \in \mathcal{H}_t, s \in \mathcal{S} \qquad \text{(D19)}$$

$$P_{kths} \geq 0, w_{kths} \in \mathbb{R}, w'_{kths} \in \mathbb{R}, y_{kths} \in \mathbb{Z} \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, h \in \mathcal{H}_t, s \in \mathcal{S} \qquad \text{(D20)}$$

$$C_{kts}, \Delta_{kts} \geq 0 \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, s \in \mathcal{S} \qquad \text{(D21)}$$

$$x_{kits} \in \{0,1\} \quad \forall k \in \mathcal{K}, i \in \mathcal{I}_k, t \in \mathcal{T}, s \in \mathcal{S} \qquad \text{(D22)}$$

NACs (8a) - (8d)

where the objective is to minimize the expectation of net expenditure during the planning period. Constraints (D1)-(D5) represents capacity balance taking into account the lifetime of each technology. Constraints (D6)-(D8) enforce expansion/budget restrictions. The inventory balance is

represented by constraints (D9) and (D10). Constraints (D11) ensure that sufficient capacity is available at all times based on the desired capacity margin. Equations (D12)-(D16) represent the up- and down-time constraints. The bounds on power generation based on the maximum/minimum power output and the time-varying availability of installed capacity is captured in constraints (D17) and (D18). Constraints (D19)-(D22) enforce bounds on the variables involved in the formulation.