A Distributed Algorithm for Multi-scale Multi-stage Stochastic Programs with Application to Electricity Capacity Expansion

A Preprint

Run Chen
School of Industrial Engineering
Purdue University
West Lafayette, IN 47906
chen885@purdue.edu

Andrew L. Liu
School of Industrial Engineering
Purdue University
West Lafayette, IN 47906
andrewliu@purdue.edu

February 22, 2023

1 Introduction

1.1 Multi-stage Stochastic Program

Multi-stage stochastic programs (MSPs) represent a framework for sequential decision making under uncertainties, where a sequence of uncertain data $(\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_T)$ is revealed gradually over stages. The decision variables $\mathbf{x}_t$, corresponding to each stage for $t = 1, \dots, T$, should be made adaptive to this process. We assume that the sequence $(\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_T)$ evolves according to a known stochastic process, and can be regarded as a sequence of random variables with a specified probability distribution. In each stage t, we use $\boldsymbol{\xi}_t$ to denote the random data vector and ξ_t to denote a specific realization; similarly, we use $\boldsymbol{\xi}_{[t]} := (\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_t)$ to denote the history of the random sequence up to stage t and $\xi_{[t]} := (\xi_1, \dots, \xi_t)$ to denote a specific realization. The decision dynamics is as follows: in each stage t, a realization $\xi_t \in \mathbb{R}^{d_t}$ is observed and followed by a decision variable $\mathbf{x}_t$, whose value, depending on ξ_t and $\mathbf{x}_{t-1}$ from previous stage, is determined to minimize the objective function value of the current stage and the expected value of future stages. A T-stage stochastic program problem can then be expressed into the following nested formulation:

$$\min_{\mathbf{x}_1 \in \mathcal{D}_1} f_1(\mathbf{x}_1) + \mathbb{E}^{\boldsymbol{\xi}_2}\big[\min_{\mathbf{x}_2 \in \mathcal{X}_2^{\boldsymbol{\xi}_2}(\mathbf{x}_1)} f_2^{\boldsymbol{\xi}_2}(\mathbf{x}_2) + \mathbb{E}^{\boldsymbol{\xi}_3|\boldsymbol{\xi}_{[2]}}\big[\cdots + \mathbb{E}^{\boldsymbol{\xi}_T|\boldsymbol{\xi}_{[T-1]}}\big[\min_{\mathbf{x}_T \in \mathcal{X}_T^{\boldsymbol{\xi}_T}(\mathbf{x}_{T-1})} f_T^{\boldsymbol{\xi}_T}(\mathbf{x}_T)\big]\big]\big], \qquad (1)$$

where $\mathbf{x}_t \in \mathbb{R}^{n_t}$ is the decision variable for each stage $t = 1, \dots, T$. $f_1 : \mathbb{R}^{n_1} \to \mathbb{R}$ is a continuous function, and $f_t^{\boldsymbol{\xi}_t} : \mathbb{R}^{n_t} \to \mathbb{R}$ is a continuous convex function depending on $\boldsymbol{\xi}_t$ for $t = 2, \dots, T$. $\mathcal{D}_1 \subset \mathbb{R}^{n_1}$ is a deterministic convex set, and the abstract constraint set $\mathcal{X}_t^{\boldsymbol{\xi}_t}(\mathbf{x}_{t-1})$ for $t = 2, \dots, T$ can be written out explicitly, for example, as a linear equality constraint:

$$\mathcal{X}_t^{\boldsymbol{\xi}_t}(\mathbf{x}_{t-1}) = \{\mathbf{x}_t \in \mathcal{D}_t | A_t^{\boldsymbol{\xi}_t}\mathbf{x}_{t-1} + B_t^{\boldsymbol{\xi}_t}\mathbf{x}_t = \mathbf{c}_t^{\boldsymbol{\xi}_t}\},$$

where $\mathcal{D}_t \subset \mathbb{R}^{n_t}$ is a deterministic convex set, $A_t^{\boldsymbol{\xi}_t} \in \mathbb{R}^{m_t \times n_{t-1}}$ and $B_t^{\boldsymbol{\xi}_t} \in \mathbb{R}^{m_t \times n_t}$ are matrices depending on $\boldsymbol{\xi}_t$, and $\mathbf{c}_t^{\boldsymbol{\xi}_t} \in \mathbb{R}^{m_t}$ is a vector depending on $\boldsymbol{\xi}_t$. Note that the explicit form of $\mathcal{X}_t^{\boldsymbol{\xi}_t}(\mathbf{x}_{t-1})$ is certainly not limited to linear equality constraints. However, as a starting point, and for the ease of presenting the detailed design and implementation of our algorithm, throughout this chapter, we only consider linearly constrained MSPs. The notation $\mathbb{E}^{\boldsymbol{\xi}_t|\boldsymbol{\xi}_{[t-1]}}$ denotes the conditional expectation operation with respect to $\boldsymbol{\xi}_t$ given $\boldsymbol{\xi}_{[t-1]}$ for

$t = 2, \ldots, T$. Note that ξ_1 is observed at the very beginning of the decision process and thus is regarded as a deterministic data, so the conditional expectation $\mathbb{E}^{\boldsymbol{\xi}_2|\boldsymbol{\xi}_{[1]}}[\cdots]$ is simply equivalent to $\mathbb{E}^{\boldsymbol{\xi}_2}[\cdots]$.

1.2 Scenario Tree and Node Separability

Computational approach for solving MSPs (1) usually is based on a discretization of the underlying stochastic process, or more specifically, an approximation with a finite number of realizations, which can be depicted in the form of a *scenario tree* [8]. Such an approximation may be constructed by various methods [7, 4, 2, 5].

Let $\mathcal{T}$ denote the set of all *nodes* in a scenario tree associated with a discretized stochastic process $(\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_T)$. The tree has a set of a finite number of nodes organized at each *level* corresponding to each stage for all $t = 1, \ldots, T$, denoted as $\mathcal{I}_t$. At level 1, there is only one *root node*, associated with the deterministic data ξ_1. From level 2, there are as many nodes as many different observations of the random data $\boldsymbol{\xi}_t$ that may occur after each realization of $\boldsymbol{\xi}_{t-1}$ from the previous level. The tree is grown recursively in such a way up to level T. Let $t(i)$ denote the level containing each node $i \in \mathcal{T}$, and the total number of nodes, denoted by N, is finite.

Given a node i and its *child node* i_c, an *edge* connecting them is associated with a probability p_{i,i_c}, denoting the conditional probability of an observation of the random data $\boldsymbol{\xi}_{t(i_c)}$ at node i_c, given a realization of $\boldsymbol{\xi}_{[t(i)]}$ at node i. Let p_i^{nd} denote the probability of each node for all $i = 1, \ldots, N$. Given that $p_1^{nd} = 1$, we obtain $p_i^{nd} = p_{a(i)}^{nd} \times p_{a(i),i}$ for $i = 2, \ldots, N$, where $a(i)$ denotes the unique *ancestor node*. Take a 4-stage stochastic process as an example, which is depicted in a scenario tree illustrated in Figure 1. There are 16 nodes in the 4-level tree and the probability of each node is calculated and shown in the figure.

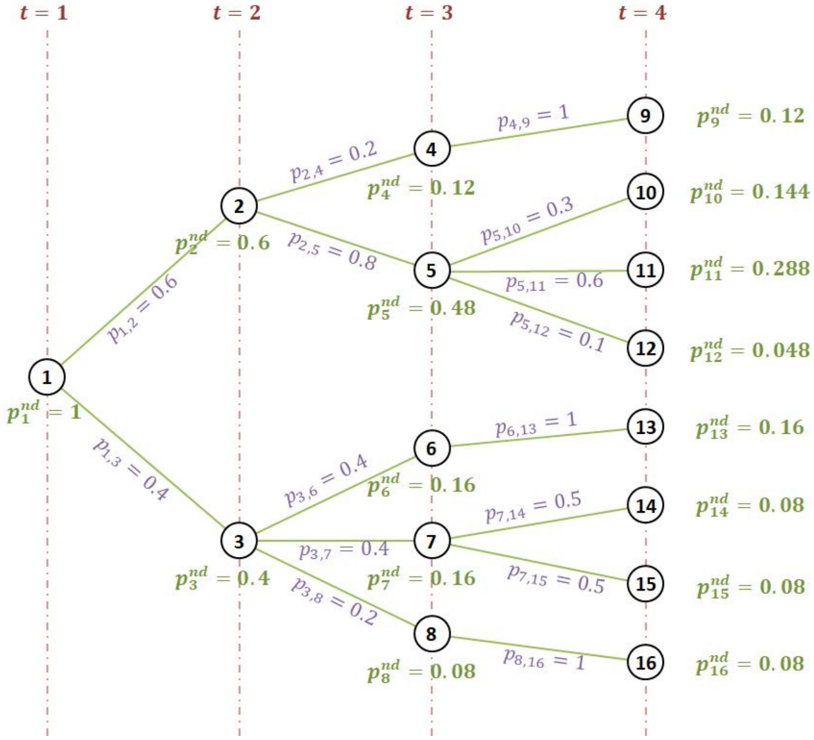

Figure 1: An example of a 4-stage stochastic process depicted in the form of a 4-level scenario tree.

A T-stage stochastic program of form (1) can then be reformulated as the following deterministic, large-scale, constrained convex optimization problem with **Node Separability**:

$$\begin{aligned} \underset{\mathbf{x}_1,\ldots,\mathbf{x}_N}{\text{minimize}} \quad & \sum_{i=1}^{N} p_i^{nd} f_i(\mathbf{x}_i) \\ \text{subject to} \quad & \mathbf{x}_i \in \mathcal{D}_{t(i)}, \quad i = 1,\ldots,N, \\ & A_i \mathbf{x}_{a(i)} + B_i \mathbf{x}_i = \mathbf{c}_i, \quad i = 2,\ldots,N, \end{aligned} \tag{2}$$

where $\mathbf{x}_i \in \mathbb{R}^{n_{t(i)}}$ is the decision variable associated with each node $i = 1,\ldots,N$. $f_i : \mathbb{R}^{n_{t(i)}} \to \mathbb{R}$ is a continuous convex function for all $i = 1,\ldots,N$. $A_i \in \mathbb{R}^{m_{t(i)} \times n_{t(a(i))}}$ and $B_i \in \mathbb{R}^{m_{t(i)} \times n_{t(i)}}$ are deterministic matrices for $i = 2,\ldots,N$, and $\mathbf{c}_i \in \mathbb{R}^{m_{t(i)}}$ is a deterministic vector for $i = 2,\ldots,N$.

1.3 Nonanticipativity and Scenario Separability

At the final level T of a scenario tree, each node $i \in \mathcal{I}_T$ is called a *leaf node*. A *scenario* is a path from the root node to a leaf node. Let $\mathcal{S}$ denote the set of all scenarios, and the total number of scenarios, denoted by S, is finite. Each scenario $s \in \mathcal{S}$ represents a possible realization of the entire stochastic process $(\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_T)$. The probability of each scenario, denoted by p_s^{sc}, is the multiplication of probabilities associated with arcs visited along the path from the root node to each leaf node. For example in Figure 1, there are 8 scenarios in total, corresponding to the 8 paths from the root node 1 to leaf nodes $i = 9,\ldots,16$. The probability of each scenario is the same as of each leaf node.

Given a finite number of scenarios, we can regard the whole decision sequence $(\mathbf{x}_1,\ldots,\mathbf{x}_T)$ as a mapping of each scenario to the whole decision space $\mathbb{R}^{n_{seq}}$, where $n_{seq} = \sum_{t=1}^{T} n_t$; that is, assigning to each scenario a vector $\mathbf{x}^s := \begin{pmatrix} \mathbf{x}_1^s \\ \vdots \\ \mathbf{x}_T^s \end{pmatrix} \in \mathbb{R}^{n_{seq}}$ for all $s = 1,\ldots,S$, where $\mathbf{x}_t^s \in \mathbb{R}^{n_t}$ for all $t = 1,\ldots,T$. Accordingly, the objective function for each scenario $f^s : \mathbb{R}^{n_{seq}} \to \mathbb{R}$ is defined as $f^s(\mathbf{x}^s) := \sum_{t=1}^{T} f_t^s(\mathbf{x}_t^s)$ for all $s = 1,\ldots,S$, where $f_t^s : \mathbb{R}^{n_t} \to \mathbb{R}$ is a continuous convex function for all $t = 1,\ldots,T$. For each scenario $s = 1,\ldots,S$, the constraint set of the decision sequence $\mathbf{x}^s$ can be denoted by

$$\mathcal{X}^s := \Big\{ \mathbf{x}^s \in \prod_{t=1}^{T} \mathcal{D}_t \Big| A_t^s \mathbf{x}_{t-1}^s + B_t^s \mathbf{x}_t = \mathbf{c}_t^s, \quad t = 2,\ldots,T \Big\},$$

where $A_t^s \in \mathbb{R}^{m_t \times n_{t-1}}$ and $B_t^s \in \mathbb{R}^{m_t \times n_t}$ are deterministic matrices for $t = 2,\ldots,T$, and $\mathbf{c}_t^s \in \mathbb{R}^{m_t}$ is a deterministic vector for $t = 2,\ldots,T$.

The decision sequence $\mathbf{x}^s$ for each scenario $s = 1,\ldots,S$ should also satisfy the *nonanticipativity constraints*. At each stage t, the scenario set $\mathcal{S}$ can be partitioned into finitely many disjoint subsets, where each scenario is observationally indistinguishable, based on the realization up to stage t. Such a subset is called a *scenario bundle* and denoted by $\mathcal{V}$. The set containing all scenario bundles at stage t is denoted by $\mathcal{U}_t$ for all $t = 1,\ldots,T$. The nonanticipativity constraints basically require $\mathbf{x}_t^s = \mathbf{x}_t^{s'}$, given any two scenarios $s, s' \in \mathcal{V}$ and any scenario bundle $\mathcal{V} \in \mathcal{U}_t$ for all $t = 1,\ldots,T$. As illustrated in Figure 2, there are 8 decision sequences assigned to the 8 scenarios of the 4-level scenario tree in Figure 1. The vertical dotted lines represent the nonaticipativity constraints. For example, at stage $t = 3, \mathcal{U}_3 = \big\{\{1\},\{2,3,4\},\{5\},\{6,7\},\{8\}\big\}$; hence we have the nonantipativity constraints: $\mathbf{x}_3^2 = \mathbf{x}_3^3 = \mathbf{x}_3^4$ and $\mathbf{x}_3^6 = \mathbf{x}_3^7$.

Let a vector $\vec{\mathbf{x}} \in \mathbb{R}^{n_{sc}}$ denote decision sequences of all scenarios: $\vec{\mathbf{x}} := \begin{pmatrix} \mathbf{x}^1 \\ \vdots \\ \mathbf{x}^S \end{pmatrix}$, and $n_{sc} = S \times n_{seq}$. Let $\mathcal{N}$ denote a subspace containing the vector $\vec{\mathbf{x}}$, where the nonanticipativity constraints are satisfied:

$$\mathcal{N} := \Big\{ \vec{\mathbf{x}} \in \mathbb{R}^{n_{sc}} \Big| \mathbf{x}_t^s = \mathbf{x}_t^{s'}, \quad \forall s, s' \in \mathcal{V}, \quad \forall \mathcal{V} \in \mathcal{U}_t, \quad t = 1,\ldots,T \Big\}.$$

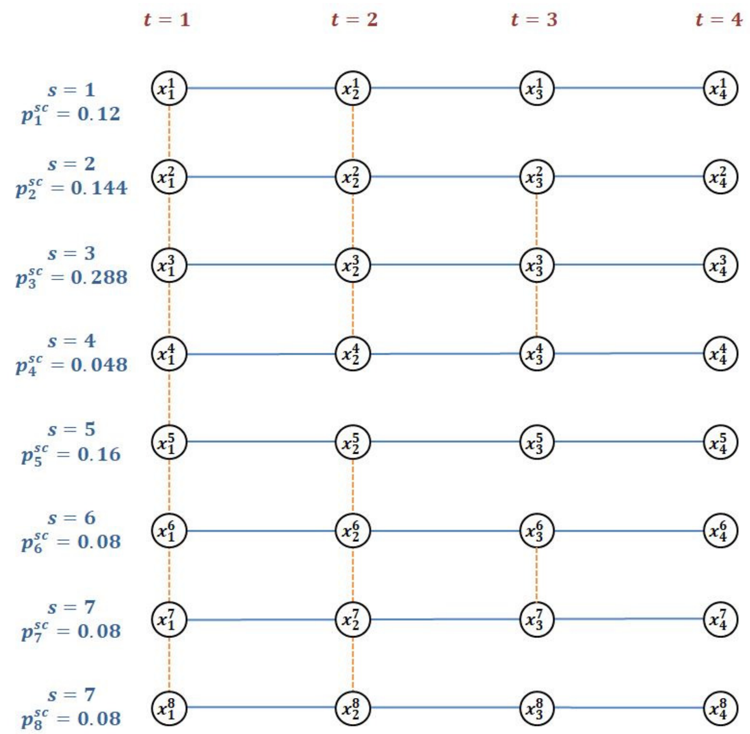
$t = 1$
$t = 2$
$t = 3$
$t = 4$
$s = 1$
$p_1^{sc} = 0.12$
x_1^1 x_2^1 x_3^1 x_4^1
$s = 2$
$p_2^{sc} = 0.144$
x_1^2 x_2^2 x_3^2 x_4^2
$s = 3$
$p_3^{sc} = 0.288$
x_1^3 x_2^3 x_3^3 x_4^3
$s = 4$
$p_4^{sc} = 0.048$
x_1^4 x_2^4 x_3^4 x_4^4
$s = 5$
$p_5^{sc} = 0.16$
x_1^5 x_2^5 x_3^5 x_4^5
$s = 6$
$p_6^{sc} = 0.08$
x_1^6 x_2^6 x_3^6 x_4^6
$s = 7$
$p_7^{sc} = 0.08$
x_1^7 x_2^7 x_3^7 x_4^7
$s = 7$
$p_8^{sc} = 0.08$
x_1^8 x_2^8 x_3^8 x_4^8

Figure 2: The decision sequence for each scenario of the 4-level scenario tree in Figure 1. Vertical dotted lines represent the nonanticipativity constraints.

Let J denote an operator that maps an arbitrary vector $\vec{\mathbf{x}} \in \mathbb{R}^{n_{sc}}$ to a unique vector $\vec{\mathbf{v}} := J\vec{\mathbf{x}}$, which is of the same dimension as $\vec{\mathbf{x}}$ and can be calculated as:

$$\mathbf{v}_t^s = \frac{\sum_{s \in \mathcal{V}} \mathbf{x}_t^s}{|\mathcal{V}|}, \quad \forall s \in \mathcal{V}, \quad \forall \mathcal{V} \in \mathcal{U}_t, \quad t = 1, \ldots, T,$$

where $|\mathcal{V}|$ denotes the total number of elements in the set $\mathcal{V}$. For example, in Figure 2, take $\mathcal{V} = \{2, 3, 4\} \in \mathcal{U}_3$ at stage $t = 3$, and $|\mathcal{V}| = 3$. Clearly, $\vec{\mathbf{v}} \in \mathcal{N}$, and the linear operator J is called an *aggregation operator* that maps from the space $\mathbb{R}^{n_{sc}}$ to the subspace $\mathcal{N}$. The nonanticipativity constraints $\vec{\mathbf{x}} \in \mathcal{N}$ can then be reformulated as $I\vec{\mathbf{x}} = J\vec{\mathbf{x}}$, where I denotes the identity operator. Denoting $K := I - J$, the nonanticipativity constraints can be further rewritten as: $K\vec{\mathbf{x}} = \mathbf{0}$, and the linear operator K can be represented using a matrix $(K_1 \cdots K_S)$ where $K_s \in \mathbb{R}^{n_{sc} \times n_{seq}}$ for all $s = 1, \ldots, S$.

A T-stage stochastic program of form (1) can be reformulated as the following deterministic, large-scale, constrained convex optimization problem with **Scenario Separability**:

$$\begin{aligned} \underset{\mathbf{x}^1, \ldots, \mathbf{x}^S}{\text{minimize}} \quad & \sum_{s=1}^{S} p_s^{sc} f^s(\mathbf{x}^s) \\ \text{subject to} \quad & \mathbf{x}^s \in \mathcal{X}^s, \quad s = 1, \ldots, S, \\ & \sum_{s=1}^{S} K_s \mathbf{x}^s = \mathbf{0}. \end{aligned} \tag{3}$$

1.4 Comparison of Node Decomposition and Scenario Decomposition

In the previous two subsections, we have showed the reformulated form of an MSP with node separability (2) and scenario separability (3) respectively. In both of the two formulations, the decision variables can be separated into multiple blocks, and the entire large-scale problem can be decomposed into multiple subproblems to be solved in parallel using a distributed algorithm. For example, PHA [6] is widely used as a scenario decomposition algorithm.

In this subsection, we make a comparison between the possible decomposition methods based on the two different types of separability. As listed in Table 1, in the node decomposition method, the large-scale problem can be decomposed all the way down by each decision variable associated with a node; while in the scenario decomposition method, it can only be decomposed down by each decision sequence assigned to a scenario, which has a much larger size that might cause memory storage issues in big-data applications. However, the explicit nonantipativity constraints make the decision sequences decoupled from the stage linking constraints, which in return brings a notable benefit.

Table 1: Comparison of Node and Scenario Decomposition

	Node Decomposition	**Scenario Decomposition**
Decompose By	node	scenario
Smallest Decomposed Unit	$\mathbf{x}^i$	$\mathbf{x}^s$
Size of Smallest Decomposed Unit	$n_{t(i)}$ **(pro)**	$n_{seq} = \sum_{t=1}^{T} n_t$ **(con)**
Number of Decomposed Unit	$N = \mathcal{O}(e^T)$	$S = \mathcal{O}(e^T)$
Coupled with Stage Linking Constraint?	yes **(con)**	no **(pro)**
Type of Nonanticipativity Constraint	implicit	explicit

2 A Hybrid Decomposition Method for Multi-scale Multi-stage Stochastic Program under Multi-scale Uncertainties

2.1 Additional Structures

In many real-world applications, the decision-making process can be mathematically modeled as an MSP in the formulation of (1). Additionally, it may have other structures that can lead to efficient algorithm design, such as the ones listed below.

(i) the decision vector $\mathbf{x}_t$ can be separated into two sub-vectors as $\mathbf{x}_t := \begin{pmatrix} \mathbf{y}_t \\ \mathbf{z}_t \end{pmatrix}$ for all $t = 1, \ldots, T$, where the sub-vector $\mathbf{y}_t \in \mathbb{R}^{n_t^Y}$ is called *aggregate level decision*, the sub-vector $\mathbf{z}_t \in \mathbb{R}^{n_t^Z}$ is called *detailed level decision*, and $n_t = n_t^Y + n_t^Z$,

(ii) the feasible region $\mathcal{D}_t$ can be separated as $\mathcal{D}_t = \mathcal{D}_t^Y \times \mathcal{D}_t^Z$ for all $t = 1, \ldots, T$, where $\mathcal{D}_t^Y \subset \mathbb{R}^{n_t^Y}$ and $\mathcal{D}_t^Z \subset \mathbb{R}^{n_t^Z}$ are convex sets,

(iii) the objective function $f_1(\mathbf{x}_1)$ can be separated as $f_1(\mathbf{x}_1) = f_1^Y(\mathbf{y}_1) + f_1^Z(\mathbf{z}_1)$, where $f_1^Y : \mathbb{R}^{n_1^Y} \to \mathbb{R}$ and $f_1^Z : \mathbb{R}^{n_1^Z} \to \mathbb{R}$ are continuous convex functions, and the objective function $f_t^{\boldsymbol{\xi}_t}(\mathbf{x}_t)$ can be separated as $f_t^{\boldsymbol{\xi}_t}(\mathbf{x}_t^s) = f_t^{Y,\boldsymbol{\xi}_t}(\mathbf{y}_t) + f_t^{Z,\boldsymbol{\xi}_t}(\mathbf{z}_t)$ for $t = 2, \ldots, T$, where $f_t^{Y,\boldsymbol{\xi}_t} : \mathbb{R}^{n_t^Y} \to \mathbb{R}$ and $f_t^{Z,\boldsymbol{\xi}_t} : \mathbb{R}^{n_t^Z} \to \mathbb{R}$ are continuous convex functions depending on $\boldsymbol{\xi}_t$,

(iv) the linear equality constraint $A_t^{\boldsymbol{\xi}_t}\mathbf{x}_{t-1} + B_t^{\boldsymbol{\xi}_t}\mathbf{x}_t = \mathbf{c}_t^{\boldsymbol{\xi}_t}$ can be rewritten as

$$\begin{pmatrix} A_t^{Y,\boldsymbol{\xi}_t} & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \mathbf{y}_{t-1} \\ \mathbf{z}_{t-1} \end{pmatrix} + \begin{pmatrix} B_t^{Y,\boldsymbol{\xi}_t} & 0 \\ B_t^{LY,\boldsymbol{\xi}_t} & B_t^{LZ,\boldsymbol{\xi}_t} \end{pmatrix}\begin{pmatrix} \mathbf{y}_t \\ \mathbf{z}_t \end{pmatrix} = \begin{pmatrix} \mathbf{c}_t^{Y,\boldsymbol{\xi}_t} \\ \mathbf{c}_t^{L,\boldsymbol{\xi}_t} \end{pmatrix}$$

for $t = 2, \dots, T$, where $A_t^{Y,\boldsymbol{\xi}_t} \in \mathbb{R}^{m_t^Y \times n_{t-1}^Y}$, $B_t^{Y,\boldsymbol{\xi}_t} \in \mathbb{R}^{m_t^Y \times n_t^Y}$, $B_t^{LY,\boldsymbol{\xi}_t} \in \mathbb{R}^{m_t^L \times n_t^Y}$ and $B_t^{LZ,\boldsymbol{\xi}_t} \in \mathbb{R}^{m_t^L \times n_t^Z}$ are matrices depending on $\boldsymbol{\xi}_t$, and $\mathbf{c}_t^{Y,\boldsymbol{\xi}_t} \in \mathbb{R}^{m_t^Y}$ and $\mathbf{c}_t^{L,\boldsymbol{\xi}_t} \in \mathbb{R}^{m_t^L}$ are vectors depending on $\boldsymbol{\xi}_t$.

In such a structure, it is observed that only the aggregated level decision $\mathbf{y}_t$'s are coupled in the stage linking constraint: $A_t^{Y,\boldsymbol{\xi}_t}\mathbf{y}_{t-1} + B_t^{Y,\boldsymbol{\xi}_t}\mathbf{y}_t = \mathbf{c}_t^{Y,\boldsymbol{\xi}_t}$; while each detailed level decision $\mathbf{z}_t$ is stagewise independent but coupled with $\mathbf{y}_t$ at the same stage through a linking constraint $B_t^{LY,\boldsymbol{\xi}_t}\mathbf{y}_t + B_t^{LZ,\boldsymbol{\xi}_t}\mathbf{z}_t = \mathbf{c}_t^{L,\boldsymbol{\xi}_t}$.

2.2 Extended Nonanticipativity and Hybrid Scenario-node Decomposition

Using the similar ways we show in Section 1, such an MSP can certainly be reformulated into a deterministic, large-scale, constrained convex optimization problem with either node or scenario separability, which has its own strength and shortcoming listed in Table 1. To reap the benefits of both methods, while overcoming their drawbacks, we propose a novel reformulation where the $\mathbf{y}$-part is with scenario separability and the $\mathbf{z}$-part is with node separability.

To each scenario of a scenario tree, we assign a decision sequence $\mathbf{y}^s := \begin{pmatrix} \mathbf{y}_1^s \\ \vdots \\ \mathbf{y}_T^s \end{pmatrix} \in \mathbb{R}^{n_{seq}^Y}$ for all $s = 1, \dots, S$, and $n_{seq}^Y = \sum_{t=1}^T n_t^Y$. Accordingly, an objective function for each scenario $f^{Y,s} : \mathbb{R}^{n_{seq}^Y} \to \mathbb{R}$ is defined as $f^{Y,s}(\mathbf{y}^s) := \sum_{t=1}^T f_t^{Y,s}(\mathbf{y}_t^s)$ for all $s = 1, \dots, S$, where $f_t^{Y,s} : \mathbb{R}^{n_t^Y} \to \mathbb{R}$ is a continuous convex function for all $t = 1, \dots, T$. For each scenario $s = 1, \dots, S$, the constraint set of the decision sequence $\mathbf{y}^s$ is denoted by

$$\mathcal{Y}^s := \Big\{\mathbf{y}^s \in \prod_{t=1}^{T} \mathcal{D}_t^Y \Big| A_t^{Y,s}\mathbf{y}_{t-1}^s + B_t^{Y,s}\mathbf{y}_t = \mathbf{c}_t^{y,s}, \quad t = 2, \dots, T\Big\},$$

where $A_t^{Y,s} \in \mathbb{R}^{m_t \times n_{t-1}}$ and $B_t^{Y,s} \in \mathbb{R}^{m_t \times n_t}$ are deterministic matrices for $t = 2, \dots, T$, and $\mathbf{c}_t^{Y,s} \in \mathbb{R}^{m_t}$ is a deterministic vector for $t = 2, \dots, T$.

With each node $i = 1, \dots, N$ in the scenario tree, a decision variable $\mathbf{z}_i \in \mathbb{R}^{n_{t(i)}^Z}$ and a local copy $\mathbf{y}_i^{nd} \in \mathbb{R}^{n_{t(i)}^Y}$ are associated, as well as a continuous convex objective function $f_i^Z : \mathbb{R}^{n_{t(i)}^Z} \to \mathbb{R}$. The decision variables $\mathbf{z}_i$ and $\mathbf{y}_i^{nd}$ are denoted together by a vector $\mathbf{w}_i^{nd} := \begin{pmatrix} \mathbf{y}_i^{nd} \\ \mathbf{z}_i \end{pmatrix} \in \mathbb{R}^{n_{t(i)}}$. The original linkage between the aggregated level decision $\mathbf{y}_t$ and the detailed level decision $\mathbf{z}_t$ can then be absorbed in the single constraint set:

$$\mathcal{W}_i^{nd} := \{\mathbf{y}_i^{nd} \in \mathbb{R}^{n_{t(i)}^Y}, \mathbf{z}_i \in \mathcal{D}_{t(i)}^Z | B_i^{LY}\mathbf{y}_i^{nd} + B_i^{LZ}\mathbf{z}_i = \mathbf{c}_i^L\}$$

for all $i = 1, \dots, N$, where $B_i^{LY} \in \mathbb{R}^{m_{t(i)}^L \times n_{t(i)}^Y}$ and $B_i^{LZ} \in \mathbb{R}^{m_{t(i)}^L \times n_{t(i)}^Z}$ are deterministic matrices, and $\mathbf{c}_i^L \in \mathbb{R}^{m_{t(i)}^L}$ is a deterministic vector. As a trade-off, the local copy $\mathbf{y}_i^{nd}$ associated with each node is additionally required to be equal to the value of $\mathbf{y}_t^s$'s in a corresponding scenario bundle.

Let us consider the 4-level scenario tree in Figure 1 again. As illustrated in Figure 3, at each stage t, each scenario bundle $\mathcal{V} \in \mathcal{U}_t$ is now vertically extended to a corresponding node i at the same level. For example, the scenario bundle $\mathcal{V} = \{6, 7\} \in \mathcal{U}_3$ at stage 3 is extended to a hybrid bundle $\mathcal{V}^{sn} = \{s = 6, s = 7, n = 7\}$. Instead of simply requiring $\mathbf{y}_3^6 = \mathbf{y}_3^7$, we now require $\mathbf{y}_3^6 = \mathbf{y}_3^7 = \mathbf{y}_7^{nd}$. We call such a hybrid bundle a *scenario-node bundle*, denoted by $\mathcal{V}^{sn}$, and the set containing all scenario-node bundles at stage t is denoted by $\mathcal{U}_t^{sn}$ for all $t = 1, \dots, T$.

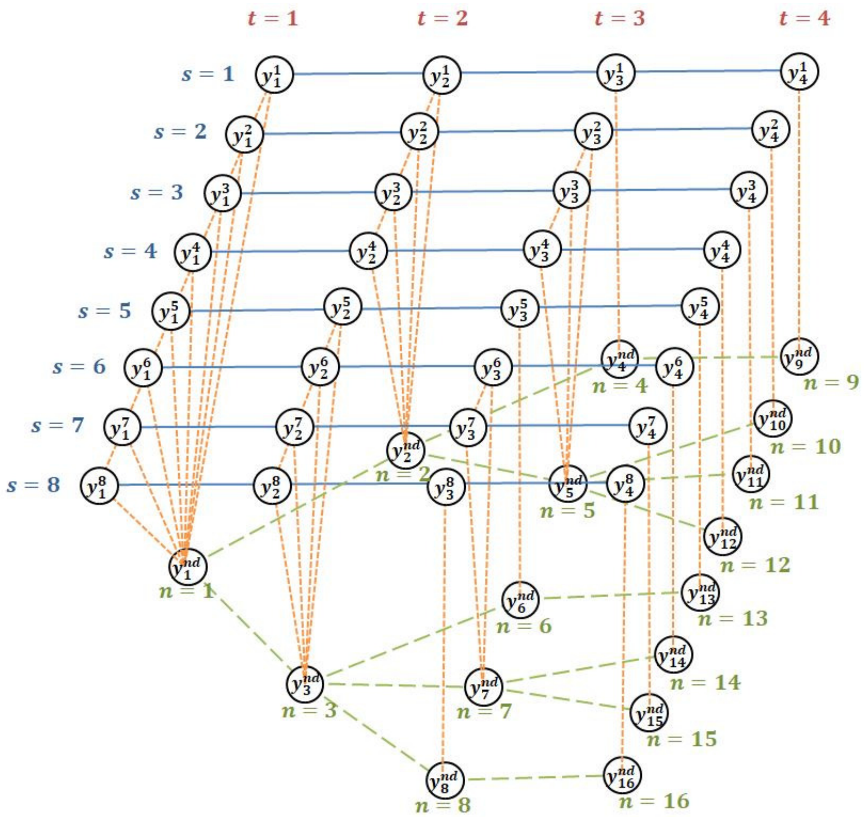

Figure 3: The decision sequence for each scenario and the local copy at each node of the 4-level scenario tree in Figure 1. Vertical dotted lines represent the extended nonanticipativity constraints.

Let a vector $\vec{\mathbf{y}}^{sn} \in \mathbb{R}^{n^Y_{sc}+n^Y_{nd}}$ denote all $\mathbf{y}$-variables: $\vec{\mathbf{y}}^{sn} := \begin{pmatrix} \mathbf{y}^1 \\ \vdots \\ \mathbf{y}^S \\ \mathbf{y}^{nd}_1 \\ \vdots \\ \mathbf{y}^{nd}_N \end{pmatrix}$, and $n^Y_{sc} = S \times n^Y_{seq}$ and $n^Y_{nd} = \sum_{i=1}^N n^Y_{t(i)}$. Let $\mathcal{N}^{sn}$ denote a subspace containing the vector $\vec{\mathbf{y}}^{sn}$, where the extended nonanticipativity constraints are satisfied; namely,

$$\mathcal{N}^{sn} := \left\{ \vec{\mathbf{y}}^{sn} \in \mathbb{R}^{n^Y_{sc}+n^Y_{nd}} \middle| \begin{array}{ll} \mathbf{y}^s_t = \mathbf{y}^{s'}_t, & \forall s, s' \in \mathcal{V}^{sn}, \\ \mathbf{y}^s_t = \mathbf{y}^{nd}_i, & \forall s, i \in \mathcal{V}^{sn}, \end{array} \quad \forall \mathcal{V}^{sn} \in \mathcal{U}^{sn}_t, \quad t = 1, \ldots, T \right\}.$$

Let J^{sn} denote the aggregation operator that maps from the space $\mathbb{R}^{n^Y_{sc}+n^Y_{nd}}$ to the subspace $\mathcal{N}^{sn}$, and the mapped vector $\vec{\mathbf{v}}^{sn} := J^{sn}\vec{\mathbf{y}}^{sn}$, of the same dimension as $\vec{\mathbf{y}}^{sn}$, can be calculated as:

$$\begin{array}{ll} \mathbf{v}^s_t = \frac{\sum_{s \in \mathcal{V}^{sn}} \mathbf{y}^s_t + \sum_{i \in \mathcal{V}^{sn}} \mathbf{y}^{nd}_i}{|\mathcal{V}|}, \quad \forall s \in \mathcal{V}^{sn}, & \\ & \forall \mathcal{V}^{sn} \in \mathcal{U}^{sn}_t, \quad t = 1, \ldots, T. \\ \mathbf{v}^{nd}_i = \frac{\sum_{s \in \mathcal{V}^{sn}} \mathbf{y}^s_t + \sum_{i \in \mathcal{V}^{sn}} \mathbf{y}^{nd}_i}{|\mathcal{V}|}, \quad \forall i \in \mathcal{V}^{sn}, & \end{array}$$

In a similar way as in Section 1.3, the extended nonanticipativity constraints $\vec{\mathbf{y}}^{sn} \in \mathcal{N}^{sn}$ can then be reformulated as $I\vec{\mathbf{y}}^{sn} = J^{sn}\vec{\mathbf{y}}^{sn}$. Denoting $K^{sn} := I - J^{sn}$, the extended nonanticipativity constraints can be further rewritten as: $K^{sn}\vec{\mathbf{y}}^{sn} = \mathbf{0}$, and the linear operator K^{sn} can be represented using a matrix $(K_1, \ldots, K_S, K^{nd}_1, \ldots, K^{nd}_N)$ where each $K_s \in \mathbb{R}^{(n^Y_{sc}+n^Y_{nd}) \times n^Y_{seq}}$ for all $s = 1, \ldots, S$, and each $K^{nd}_i \in \mathbb{R}^{(n^Y_{sc}+n^Y_{nd}) \times n^Y_{t(i)}}$ for all $i = 1, \ldots, N$.

A T-stage stochastic program of form (1) with the additional structure stated in Section 2.1 can then be reformulated as the following deterministic, large-scale, constrained convex optimization problem with **Hybrid Scenario-node Separability**:

$$\begin{aligned} \underset{\substack{\mathbf{y}^1,\dots,\mathbf{y}^S \\ \mathbf{w}_1^{nd},\dots,\mathbf{w}_N^{nd}}}{\text{minimize}} \quad & \sum_{s=1}^{S} p_s^{sc} f^{Y,s}(\mathbf{y}^s) + \sum_{i=1}^{N} p_i^{nd} f_i^Z(\mathbf{z}_i) \\ \text{subject to} \quad & \mathbf{y}^s \in \mathcal{Y}^s, \quad s = 1, \dots, S, \\ & \mathbf{w}_i^{nd} \in \mathcal{W}_i^{nd}, \quad i = 1, \dots, N, \\ & \sum_{s=1}^{S} K_s \mathbf{y}^s + \sum_{i=1}^{N} K_i^{nd} \mathbf{y}_i^{nd} = \mathbf{0}. \end{aligned} \tag{4}$$

Compared with the node-only and scenario-only decomposition methods, the hybrid scenario-node decomposition inherits both of their strengths but none of their shortcomings, as listed in Table 2. The entire problem is decomposed all the way down by the smallest separate blocks of the decision variables, which are $\mathbf{y}^s$ for all $s = 1, \dots, S$ and $\mathbf{w}_i^{nd}$ for all $i = 1, \dots, N$. Note that the size of the aggregated level decision variable $\mathbf{y}_t$ is usually much smaller than that of the detailed level decision variable $\mathbf{z}_t$; i.e., $n_t^Y \ll n_t^Z$. For example, in an electricity capacity expansion model, $\mathbf{y}_t$ represents the yearly planning variables while $\mathbf{z}_t$ represents the hourly operational variables, whose size can be 10^3 times of $\mathbf{y}_t$. In this case, $n_{seq}^Y = \sum_{t=1}^T n_t^Y \ll n_{t(i)} = n_{t(i)}^Y + n_{t(i)}^Z$. Each decomposed unit, either $\mathbf{y}^s$ or $\mathbf{w}_i^{nd}$ is not coupled with any stage linking constraint, making the resulting deterministic problem suitable for massive parallel computing.

Table 2: Comparison of Hybrid Scenario-node Decomposition with Node-only and Scenario-only Decomposition

	Node Decomposition	**Scenario Decomposition**	**Hybrid Scenario-node Decomposition**
Decompose By	node only	scenario only	scenario and node
Smallest Decomposed Unit	$\begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix}^i$	$\begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix}^s$	$\mathbf{y}^s$ and $\mathbf{w}^i$
Size of Smallest Decomposed Unit	$n_{t(i)}$ **(pro)**	$n_{seq} = \sum_{t=1}^T n_t$ **(con)**	$n_{seq}^Y = \sum_{t=1}^T n_t^Y$ and $n_{t(i)}$ **(pro)**
Number of Decomposed Unit	$N = \mathcal{O}(e^T)$	$S = \mathcal{O}(e^T)$	$S + N = \mathcal{O}(e^T)$
Coupled with Stage Linking Constraint?	yes **(con)**	no **(pro)**	no **(pro)**
Type of Nonanticipativity Constraint	implicit	explicit	explicit and extended

2.3 Under Multi-scale Uncertainties

In many applications, not only the size of the detailed level decision variable is huge, but also various uncertainties of different temporal scales are involved. At each node of a scenario tree, assume that the decision variable $\mathbf{z}_i$ additionally depends on an uncertainty $\boldsymbol{\eta}_i$ of different temporal scale from $\boldsymbol{\xi}_t$ for all $i = 1, \dots, N$, which is further assumed to have a finite number of realizations $r = 1, \dots, R_i$ with probability p_r^{rl}.

Similarly, with each realization of $\boldsymbol{\eta}_i$ at node i, a decision variable $\mathbf{z}_{i,r} \in \mathbb{R}^{n^Z_{t(i)}}$ and a local copy $\mathbf{y}^{nr}_{i,r} \in \mathbb{R}^{n^Y_{t(i)}}$ are associated, as well as a continuous convex objective function $f^Z_{i,r} : \mathbb{R}^{n^Z_{t(i)}} \to \mathbb{R}$ for all $r = 1, \dots, R_i$ and $i = 1, \dots, N$. The decision variables $\mathbf{z}_{i,r}$ and $\mathbf{y}^{nr}_{i,r}$ are denoted together by a vector $\mathbf{w}^{nr}_{i,r} := \begin{pmatrix} \mathbf{y}^{nr}_{i,r} \\ \mathbf{z}_{i,r} \end{pmatrix} \in \mathbb{R}^{n_{t(i)}}$, subject to a constraint set:

$$\mathcal{W}^{nr}_{i,r} := \{\mathbf{y}^{nr}_{i,r} \in \mathbb{R}^{n^Y_{t(i)}}, \mathbf{z}_{i,r} \in \mathcal{D}^Z_{t(i)} | B^{LY}_{i,r}\mathbf{y}^{nr}_{i,r} + B^{LZ}_{i,r}\mathbf{z}_{i,r} = \mathbf{c}^L_{i,r}\}$$

for all $r = 1, \dots, R_i$ and $i = 1, \dots, N$, where $B^{LY}_{i,r} \in \mathbb{R}^{m^L_{t(i)} \times n^Y_{t(i)}}$ and $B^{LZ}_{i,r} \in \mathbb{R}^{m^L_{t(i)} \times n^Z_{t(i)}}$ are deterministic matrices, and $\mathbf{c}^L_{i,r} \in \mathbb{R}^{m^L_{t(i)}}$ is a deterministic vector.

Still consider the 4-level scenario tree in Figure 1, and assume that $\boldsymbol{\eta}_i$ at each node i has 4 possible realizations. As illustrated in Figure 4, the scenario bundle $\mathcal{V} = \{6, 7\} \in \mathcal{U}_3$ at stage 3 is further extended to a hybrid *scenario-node-realization bundle* $\mathcal{V}^{snr} = \{s = 6, s = 7, (i, r) = (7, 1), (i, r) = (7, 2), (i, r) = (7, 3), (i, r) = (7, 4)\}$. The extended nonanticipativity constraints become: $\mathbf{y}^6_3 = \mathbf{y}^7_3 = \mathbf{y}^{nr}_{7,1} = \mathbf{y}^{nr}_{7,2} =$

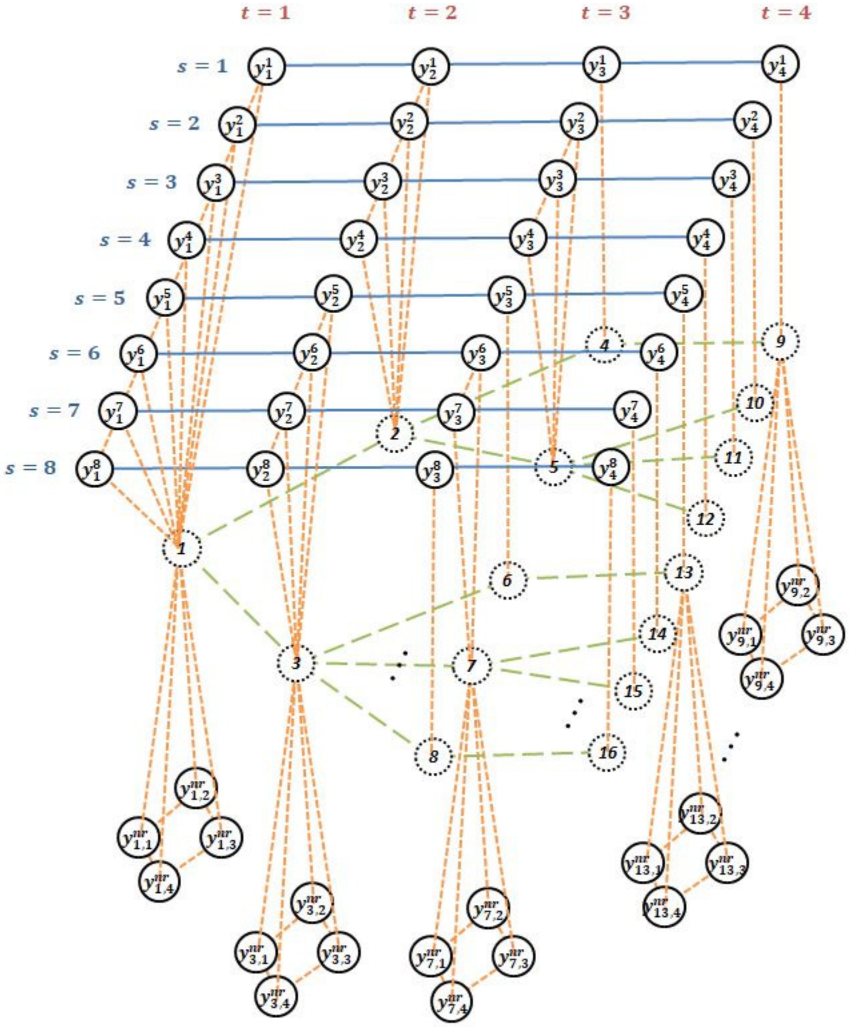

Figure 4: The decision sequence for each scenario and the local copy for each realization of the multi-scale uncertainty at each node of the 4-level scenario tree in Figure 1. Vertical dotted lines represent the extended nonanticipativity constraints.

$\mathbf{y}^{nr}_{7,3} = \mathbf{y}^{nr}_{7,4}$. The set containing all scenario-node-realization bundles at stage t is denoted by $\mathcal{U}^{snr}_t$ for all $t = 1, \ldots, T$.

Similarly, let a vector $\vec{\mathbf{y}}^{snr} \in \mathbb{R}^{n^Y_{sc}+n^Y_{nr}}$ denote all $\mathbf{y}$-variables: $\vec{\mathbf{y}}^{snr} := \begin{pmatrix} \mathbf{y}^1 \\ \vdots \\ \mathbf{y}^S \\ \mathbf{y}^{nr}_{1,1} \\ \vdots \\ \mathbf{y}^{nr}_{N,R_N} \end{pmatrix}$, and $n^Y_{sc} = S \times n^Y_{seq}$ and $n^Y_{nr} = \sum_{i=1}^N \left[R_i n^Y_{t(i)}\right]$. Let $\mathcal{N}^{snr}$ denote a subspace containing the vector $\vec{\mathbf{y}}^{snr}$ where the extended nonanticipativity constraints are satisfied as follows:

$$\mathcal{N}^{snr} := \left\{ \vec{\mathbf{y}}^{snr} \in \mathbb{R}^{n^Y_{sc}+n^Y_{nr}} \middle| \begin{array}{ll} \mathbf{y}^s_t = \mathbf{y}^{s'}_t, & \forall s, s' \in \mathcal{V}^{snr}, \\ \mathbf{y}^{nr}_{i,r} = \mathbf{y}^{nr}_{i,r'}, & \forall (i,r), (i,r') \in \mathcal{V}^{snr}, \\ \mathbf{y}^s_t = \mathbf{y}^{nr}_{i,r}, & \forall s, (i,r) \in \mathcal{V}^{snr}, \end{array} \right. \\ \left. \forall \mathcal{V}^{snr} \in \mathcal{U}^{snr}_t, \quad t = 1, \ldots, T \right\}. \tag{5}$$

Let J^{snr} denote the aggregation operator that maps from the space $\mathbb{R}^{Sn^Y_{seq}+n^Y_{nr}}$ to the subspace $\mathcal{N}^{snr}$, and the mapped vector $\vec{\mathbf{v}}^{snr} := J^{snr}\vec{\mathbf{y}}^{snr}$, of the same dimension as $\vec{\mathbf{y}}^{snr}$, can be calculated as:

$$\left.\begin{array}{ll} \mathbf{v}^s_t = \frac{\sum_{s \in \mathcal{V}^{snr}} \mathbf{y}^s_t + \sum_{(i,r) \in \mathcal{V}^{snr}} \mathbf{y}^{nr}_{i,r}}{|\mathcal{V}|}, & \forall s \in \mathcal{V}^{snr}, \\ \mathbf{v}^{nr}_{i,r} = \frac{\sum_{s \in \mathcal{V}^{snr}} \mathbf{y}^s_t + \sum_{(i,r) \in \mathcal{V}^{snr}} \mathbf{y}^{nr}_{i,r}}{|\mathcal{V}|}, & \forall (i,r) \in \mathcal{V}^{snr}, \end{array}\right. \quad \forall \mathcal{V}^{snr} \in \mathcal{U}^{snr}_t, \quad t = 1, \ldots, T.$$

In a similar way as in Section 1.3, the extended nonanticipativity constraints $\vec{\mathbf{y}}^{snr} \in \mathcal{N}^{snr}$ can then be reformulated as $I\vec{\mathbf{y}}^{snr} = J^{snr}\vec{\mathbf{y}}^{snr}$. Denoting $K^{snr} := I - J^{snr}$, the extended nonanticipativity constraints can be further rewritten as: $K^{snr}\vec{\mathbf{y}}^{snr} = \mathbf{0}$, and the linear operator K^{snr} can be represented using a matrix $(K_1, \ldots, K_S, K^{nr}_{1,1}, \ldots, K^{nr}_{N,R_N})$ where each $K_s \in \mathbb{R}^{(n^Y_{sc}+n^Y_{nr}) \times n^Y_{seq}}$ for all $s = 1, \ldots, S$, and each $K^{nr}_{i,r} \in \mathbb{R}^{(n^Y_{sc}+n^Y_{nd}) \times n^Y_{t(i)}}$ for all $r = 1, \ldots, R_i$ and $i = 1, \ldots, N$.

Under multi-scale uncertainties, a T-stage stochastic program of form (1) with the additional structure stated in Section 2.1 can be reformulated as the following deterministic, large-scale, constrained convex optimization problem with **Hybrid Scenario-node-realization Separability**:

$$\begin{array}{ll} \underset{\substack{\mathbf{y}^1, \ldots, \mathbf{y}^S \\ \mathbf{w}^{nr}_{1,1}, \ldots, \mathbf{w}^{nr}_{N,R_N}}}{\text{minimize}} & \sum_{s=1}^S p^{sc}_s f^{Y,s}(\mathbf{y}^s) + \sum_{i=1}^N p^{nd}_i \Big[\sum_{r=1}^{R_i} p^{rl}_r f^Z_{i,r}(\mathbf{z}_{i,r}) \Big] \\ \text{subject to} & \mathbf{y}^s \in \mathcal{Y}^s, \quad s = 1, \ldots, S, \\ & \mathbf{w}^{nr}_{i,r} \in \mathcal{W}^{nr}_{i,r}, \quad r = 1, \ldots, R_i, \quad i = 1, \ldots, N, \\ & \sum_{s=1}^S K_s \mathbf{y}^s + \sum_{i=1}^N \sum_{r=1}^{R_i} K^{nr}_{i,r} \mathbf{y}^{nr}_{i,r} = \mathbf{0}. \end{array} \tag{6}$$

3 A Simplified PCPM Algorithm using Orthogonal Projection

The decision variables of the optimization problem (6) can be divided into two groups: $\vec{\mathbf{y}} := \begin{pmatrix} \mathbf{y}^1 \\ \vdots \\ \mathbf{y}^S \end{pmatrix}$ and $\vec{\mathbf{w}} := \begin{pmatrix} \mathbf{w}^{nr}_{1,1} \\ \vdots \\ \mathbf{w}^{nr}_{N,R_N} \end{pmatrix}$. The vector $\vec{\mathbf{y}}$ can be further separated into blocks of $\mathbf{y}^s$ for all $s = 1, \ldots, S$, and the vector $\vec{\mathbf{w}}$ can be further separated into blocks of $\mathbf{w}_{i,r}^{nr} = \begin{pmatrix} \mathbf{y}_{i,r}^{nr} \\ \mathbf{z}_{i,r} \end{pmatrix}$ for all $r = 1, \ldots, R_i$ and $i = 1, \ldots, N$. The objective function in problem (6) is the summation of a set of separable functions defined on all blocks, coupled by the extended nonanticipativity constraints, which are linear equality constraints. We can apply the ADMM algorithm or the PCPM algorithm to solve problem (6) in a distributed fashion.

Let $\vec{\boldsymbol{\lambda}} \in \mathbb{R}^{n_{sc}^Y + n_{nr}^Y}$ denote the Lagrangian multiplier, associated with the extended nonanticipativity constraint, in the form of $\vec{\boldsymbol{\lambda}} = \begin{pmatrix} \boldsymbol{\lambda}^1 \\ \vdots \\ \boldsymbol{\lambda}^S \\ \boldsymbol{\lambda}_{1,1}^{nr} \\ \vdots \\ \boldsymbol{\lambda}_{N,R_N}^{nr} \end{pmatrix}$. The classic Lagrangian function $\mathcal{L} : \left(\prod_{s=1}^S \mathcal{Y}^s\right) \times \left(\prod_{i=1}^N \prod_{r=1}^{R_i} \mathcal{W}_{i,r}^{nr}\right) \times \mathbb{R}^{n_{sc}^Y + n_{nr}^Y} \to \mathbb{R}$ is defined as:

$$\begin{aligned} \mathcal{L}(\vec{\mathbf{y}}, \vec{\mathbf{w}}, \vec{\boldsymbol{\lambda}}) = & \sum_{s=1}^S p_s^{sc} f^{Y,s}(\mathbf{y}^s) + \sum_{i=1}^N \sum_{r=1}^{R_i} \left[p_i^{nd} p_r^{rl} f_{i,r}^Z(\mathbf{z}_{i,r}) \right] \\ & + \vec{\boldsymbol{\lambda}}^T \Big(\sum_{s=1}^S K_s \mathbf{y}^s + \sum_{i=1}^N \sum_{r=1}^{R_i} K_{i,r}^{nr} \mathbf{y}_{i,r}^{nr} \Big), \\ & \forall \mathbf{y}^s \in \mathcal{Y}^s, \quad s = 1, \ldots, S, \\ & \forall \mathbf{w}_{i,r}^{nr} \in \mathcal{W}_{i,r}^{nr}, \quad r = 1, \ldots, R_i, \quad i = 1, \ldots, N. \end{aligned} \tag{7}$$

It is well-known that for a convex problem of the specific form in (6), finding an optimal solution is equivalent to finding a saddle point $(\vec{\mathbf{y}}^*, \vec{\mathbf{w}}^*, \vec{\boldsymbol{\lambda}}^*)$ such that $\mathcal{L}(\vec{\mathbf{y}}^*, \vec{\mathbf{w}}^*, \vec{\boldsymbol{\lambda}}) \leq \mathcal{L}(\vec{\mathbf{y}}^*, \vec{\mathbf{w}}^*, \vec{\boldsymbol{\lambda}}^*) \leq \mathcal{L}(\vec{\mathbf{y}}, \vec{\mathbf{w}}, \vec{\boldsymbol{\lambda}}^*)$. We assume that such a saddle point always exists for problem (6).

3.1 Apply the PCPM Algorithm

For the ease of argument, we write out again the deterministic equivalent formula from an MSP below, which is the same as (6).

$$\begin{aligned} \underset{\substack{\mathbf{y}^s \in \mathcal{Y}^s \\ \mathbf{w}_{i,r}^{nr} \in \mathcal{W}_{i,r}^{nr}}}{\text{minimize}} \quad & \sum_{s=1}^S p_s^{sc} f^{Y,s}(\mathbf{y}^s) + \sum_{i=1}^N \sum_{r=1}^{R_i} \left[p_i^{nd} p_r^{rl} f_{i,r}^Z(\mathbf{z}_{i,r}) \right] \\ \text{subject to} \quad & \sum_{s=1}^S K_s \mathbf{y}^s + \sum_{i=1}^N \sum_{r=1}^{R_i} K_{i,r}^{nr} \mathbf{y}_{i,r}^{nr} = \mathbf{0}. \qquad (\vec{\boldsymbol{\lambda}}) \end{aligned} \tag{8}$$

To apply the PCPM algorithm [1], at each iteration k, with a given primal-dual pair, $(\vec{\mathbf{y}}^k, \vec{\mathbf{w}}^k, \vec{\boldsymbol{\lambda}}^k)$, we start with a dual predictor update:

$$\vec{\mathbf{p}}^{k+1} := \vec{\boldsymbol{\lambda}}^k + \rho \Big(\sum_{s=1}^S K_s \mathbf{y}^{s,k} + \sum_{i=1}^N \sum_{r=1}^{R_i} K_{i,r}^{nr} \mathbf{y}_{i,r}^{nr,k} \Big). \tag{9}$$

After the dual predictor update, we update the primal variables $(\vec{\mathbf{y}}^{k+1}, \vec{\mathbf{w}}^{k+1})$ by minimizing the Lagrangian function $\mathcal{L}(\vec{\mathbf{y}}, \vec{\mathbf{w}}, \vec{\mathbf{p}}^{k+1})$ evaluated at the dual predictor variable $\vec{\mathbf{p}}^{k+1}$, plus the proximal terms. The primal minimization step can be written as

$$\mathbf{y}^{s,k+1} = \underset{\mathbf{y}^s \in \mathcal{Y}^s}{\operatorname{argmin}} \; p_s^{sc} f^{Y,s}(\mathbf{y}^s) + (\vec{\mathbf{p}}^{k+1})^T K_s \mathbf{y}^s + \frac{1}{2\rho} \|\mathbf{y}^s - \mathbf{y}^{s,k}\|_2^2, \quad s = 1, \ldots, S, \tag{10a}$$

$$\begin{aligned} \mathbf{w}_{i,r}^{nr,k+1} = \underset{\mathbf{w}_{i,r}^{nr} \in \mathcal{W}_{i,r}^{nr}}{\operatorname{argmin}} \; & p_i^{nd} p_r^{rl} f_{i,r}^Z(\mathbf{z}_{i,r}) + (\vec{\mathbf{p}}^{k+1})^T K_{i,r}^{nr} \mathbf{y}_{i,r}^{nr} + \frac{1}{2\rho} \|\mathbf{y}_{i,r}^{nr} - \mathbf{y}_{i,r}^{nr,k}\|_2^2 \\ & + \frac{1}{2\rho} \|\mathbf{z}_{i,r} - \mathbf{z}_{i,r}^k\|_2^2, \quad r = 1, \ldots, R_i, \quad i = 1, \ldots, N. \end{aligned} \tag{10b}$$

A dual corrector update is then performed for each Lagrangian multiplier:

$$\vec{\boldsymbol{\lambda}}^{k+1} = \vec{\boldsymbol{\lambda}}^{k} + \rho\Big(\sum_{s=1}^{S} K_s \mathbf{y}^{s,k+1} + \sum_{i=1}^{N}\sum_{r=1}^{R_i} K_{i,r}^{nr}\mathbf{y}_{i,r}^{nr,k+1}\Big). \tag{11}$$

3.2 Orthogonal Projection

The convergence of the PCPM algorithm to an optimal solution under proper assumptions is well analyzed in [1]. However, we do find an implementation scheme that simplifies the primal minimization step, based on a nice property of the aggregation operator J^{snr}. We first write out the well-known definition of an *orthogonal projection* in our context.

Definition 3.1 (Orthogonal Projection)**.** Given a vector space $\mathbb{V}$ equipped with an inner product and a subspace $\mathbb{W}$, consider an operator $P : \mathbb{V} \to \mathbb{W}$ that maps a vector $\mathbf{v} \in \mathbb{V}$ to a unique vector $\mathbf{u} \in \mathbb{W}$. P is called a *projection* if $P = P^2$. P is further called an *orthogonal projection* if $\mathbf{v} - P\mathbf{v} \perp \mathbb{W}$ for any $\mathbf{v} \in \mathbb{V}$, i.e., $\langle \mathbf{v} - P\mathbf{v}, \mathbf{w}\rangle = 0$ for any $\mathbf{v} \in \mathbb{V}$ and $\mathbf{w} \in \mathbb{W}$.

Then, we present the following lemma, showing a nice property of the orthogonal projection.

Lemma 3.1. Given a vector space $\mathbb{V}$ equipped with an inner product and a subspace $\mathbb{W}$, if $P : \mathbb{V} \to \mathbb{W}$ is an orthogonal projection, then we have

$$\langle P\mathbf{v}, \mathbf{v}'\rangle = \langle P\mathbf{v}, P\mathbf{v}'\rangle = \langle \mathbf{v}, P\mathbf{v}'\rangle, \quad \forall \mathbf{v}, \mathbf{v}' \in \mathbb{V}.$$

Proof. The first equality holds because $\langle P\mathbf{v}, \mathbf{v}'\rangle = \langle P\mathbf{v}, (\mathbf{v}' - P\mathbf{v}') + P\mathbf{v}'\rangle = \langle P\mathbf{v}, \mathbf{v}' - P\mathbf{v}'\rangle + \langle P\mathbf{v}, P\mathbf{v}'\rangle = \langle P\mathbf{v}, P\mathbf{v}'\rangle$. Similarly, the second equality also holds. □

It is easy to check that the aggregation operator J^{snr} is an orthogonal projection from the space $\mathbb{R}^{n_{sc}^Y + n_{nr}^Y}$ to the subspace $\mathcal{N}^{snr}$, where the extended nonanticipativity constraints are satisfied. Also, $K^{snr} = I - J^{snr}$ is an orthogonal projection from the space $\mathbb{R}^{n_{sc}^Y + n_{nr}^Y}$ to the subspace $\mathcal{N}^{snr\perp}$, where $\mathcal{N}^{snr\perp}$ denotes the subspace orthogonal to $\mathcal{N}^{snr}$.

3.3 A Simplified PCPM Algorithm

We observe that the term $\vec{\mathbf{p}}^T \left(\sum_{s=1}^{S} K_s \mathbf{y}^s + \sum_{i=1}^{N} \sum_{r=1}^{R_i} K_{i,r}^{nr}\mathbf{y}_{i,r}^{nr}\right)$ in primal minimization steps (10a) and (10b) can be written as an inner product: $\langle \vec{\mathbf{p}}, K^{snr}\vec{\mathbf{y}}^{snr}\rangle$. By Lemma 3.1, $\langle \vec{\mathbf{p}}, K^{snr}\vec{\mathbf{y}}^{snr}\rangle = \langle K^{snr}\vec{\mathbf{p}}, \vec{\mathbf{y}}^{snr}\rangle$, since K^{snr} is an orthogonal projection. Denoting $\vec{\mathbf{q}} := K^{snr}\vec{\mathbf{p}}$, the inner product $\langle \vec{\mathbf{p}}, K^{snr}\vec{\mathbf{y}}^{snr}\rangle$ can be further rewritten as $\langle \vec{\mathbf{p}}, K^{snr}\vec{\mathbf{y}}^{snr}\rangle = \langle K^{snr}\vec{\mathbf{p}}, \vec{\mathbf{y}}^{snr}\rangle = \langle \vec{\mathbf{q}}, \vec{\mathbf{y}}^{snr}\rangle = \sum_{s=1}^{S} (\mathbf{q}^s)^T \mathbf{y}^s + \sum_{i=1}^{N} \sum_{r=1}^{R_i} (\mathbf{q}_{i,r}^{nr})^T \mathbf{y}_{i,r}^{nr}$.

The dual predictor update (9) can be written in the form of

$$\vec{\mathbf{p}}^{k+1} = \vec{\boldsymbol{\lambda}}^k + \rho K^{snr}\vec{\mathbf{y}}^{snr}.$$

Applying the orthogonal projection on both sides, we obtain

$$K^{snr}\vec{\mathbf{p}}^{k+1} = K^{snr}\vec{\boldsymbol{\lambda}}^k + \rho K^{snr}K^{snr}\vec{\mathbf{y}}^{snr},$$

which is equivalent to

$$\begin{aligned} \vec{\mathbf{q}}^{k+1} =& \vec{\boldsymbol{\gamma}}^k + \rho K^{snr}\vec{\mathbf{y}}^{snr} \\ =& \vec{\boldsymbol{\gamma}}^k + \rho\Big(\sum_{s=1}^{S} K_s \mathbf{y}^{s,k} + \sum_{i=1}^{N}\sum_{r=1}^{R_i} K_{i,r}^{nr}\mathbf{y}_{i,r}^{nr,k}\Big), \end{aligned} \tag{12}$$

where $\vec{\gamma} := K^{snr}\vec{\lambda} \in \mathcal{N}^{snr\perp}$ is the new Lagrangian multiplier. The primal minimization step (10a) and (10b) can be simplified as

$$\mathbf{y}^{s,k+1} = \operatorname*{argmin}_{\mathbf{y}^s \in \mathcal{Y}^s} \; p_s^{sc} f^{Y,s}(\mathbf{y}^s) + (\mathbf{q}^{s,k+1})^T \mathbf{y}^s + \frac{1}{2\rho}\|\mathbf{y}^s - \mathbf{y}^{s,k}\|_2^2, \quad s = 1, \dots, S, \tag{13a}$$

$$\begin{aligned} \mathbf{w}_{i,r}^{nr,k+1} = \operatorname*{argmin}_{\mathbf{w}_{i,r}^{nr} \in \mathcal{W}_{i,r}^{nr}} \; & p_i^{nd} p_r^{rl} f_{i,r}^{Z}(\mathbf{z}_{i,r}) + (\mathbf{q}_{i,r}^{nr,k+1})^T \mathbf{y}_{i,r}^{nr} + \frac{1}{2\rho}\|\mathbf{y}_{i,r}^{nr} - \mathbf{y}_{i,r}^{nr,k}\|_2^2 \\ & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad y \\ & + \frac{1}{2\rho}\|\mathbf{z}_{i,r} - \mathbf{z}_{i,r}^{k}\|_2^2, \quad r = 1, \dots, R_i, \quad i = 1, \dots, N. \end{aligned} \tag{13b}$$

It can be easily observed that all the K-matrices in (10a) and (10b) no longer show up in (13a) and (13b), which greatly simplifies the primal minimization steps. Originally, the calculation of the term $\vec{\mathbf{p}}^T K_s$ or $\vec{\mathbf{p}}^T K_{i,r}^{nr}$ needs values of all components of the Lagrangian multiplier $\vec{\mathbf{p}}$, which are stored distributively, and hence requires extra communication among all computing units. Now, for the update of each $\mathbf{y}$-block or $\mathbf{w}$-block, only the value of the corresponding component $\mathbf{q}^s$ or $\mathbf{q}_{i,r}^{nr}$ is required and can be stored locally in the computing unit responsible for $\mathbf{y}^s$ or $\mathbf{w}_{i,r}^{nr}$. There is no need of any communication among all computing units for the primal minimization step (13a) and (13b). Similarly to the dual predictor update, applying the orthogonal projection on both sides of the dual corrector update (11) yields

$$\vec{\gamma}^{k+1} = \vec{\gamma}^{k} + \rho\Big(\sum_{s=1}^{S} K_s \mathbf{y}^{s,k+1} + \sum_{i=1}^{N}\sum_{r=1}^{R_i} K_{i,r}^{nr}\mathbf{y}_{i,r}^{nr,k+1}\Big). \tag{14}$$

The overall simplified structure of the PCPM algorithm is presented in Algorithm 1 below.

Algorithm 1 Simplified PCPM for solving (6)

```
Initialization choose an arbitrary starting point $(\vec{\mathbf{y}}^0, \vec{\mathbf{w}}^0, \vec{\gamma}^0)$.
$k \leftarrow 0$.
while termination conditions are not met do
    (Dual Predictor Update)
     update $\vec{\mathbf{q}}^{k+1}$ according to (12);
    (Primal Update)
     update $\vec{\mathbf{y}}^{k+1}$ and $\vec{\mathbf{w}}^{k+1}$ according to (13a) and (13b);
    (Dual Corrector Update)
     update $\vec{\gamma}^{k+1}$ according to (14);
    $k \leftarrow k + 1$
return $(\vec{\mathbf{y}}^k, \vec{\mathbf{w}}^k, \vec{\gamma}^k)$.
```

4 Electricity Capacity Expansion under Multi-scale Uncertainties

We begin this section by presenting a co-optimization model where the long-term electricity capacity expansion is co-optimized with short-term generation and transmission constraints.

4.1 Capacity Expansion Planning

At the beginning of each year $t = 1, \dots, T$, a decision of an expanded capacity $x_{g,t}$ for each power generator $g \in \mathbb{G}$ has to be made adaptive to a stochastic process, and a cumulative capacity $k_{g,t}$ for each power generator $g \in \mathbb{G}$ is aggregated as $k_{g,t} = k_{g,t-1} + x_{g,t}$ for $t = 2, \dots, T$. We regard each year t as a stage in an MSP. Given a scenario tree with a set of S scenarios, we describe the model of capacity expansion planning. All indices, sets and functions are listed in Table 3; the decision variables are listed in Table 4. The overnight investment cost of an expanded capacity is calculated using a quadratic form: $\frac{1}{2}IC_{g,t}^s(x_{g,t}^s)^2$ and is levelized

Table 3: Indices, Sets and Functions for Capacity Expansion Planning

$\mathcal{S}$	set of a finite number of scenarios, indexed s;
p_s^{sc}	probability of each scenario s;
$\mathbb{J}$	set of substations, indexed j;
$\mathbb{G}$	set of power generators, indexed g;
$\hat{\mathbb{J}}$	set of reserve margin regions, indexed $\hat{j}$;
$\mathcal{J}(j)$	function that maps a substation $j \in \mathbb{J}$ to a reserve margin region $\hat{j} \in \hat{\mathbb{J}}$;
T	number of years in the planning horizon, indexed t;

Table 4: Decision Variables for Capacity Expansion Planning

$x_{g,t}^s$	expanded capacity of power generator g in year t for scenario s; [MW]
$k_{g,t}^s$	cumulative capacity of power generator g in year t for scenario s; [MW]

through N_g installments in the future for any $g \in \mathbb{G}$ and $t = 1, \ldots, T$, where $IC_{g,t}^s$ is the quadratic coefficient for scenario $s = 1, \ldots, S$. All parameters for capacity expansion planning are listed in Table 5. There's also a fixed operation and maintenance (O&M) cost of the cumulative capacity: $FOM_g k_{g,t}^s$ for any $g \in \mathbb{G}$ and $t = 1, \ldots, T$. Then, the total cost of capacity expansion planning is:

Table 5: Parameters for Capacity Expansion Planning

δ	discount factor; $0 < \delta < 1$;
$IC_{g,t}^s$	quadratic coefficient for the overnight investment cost of power generator g in year t for scenario s; [k\$/MW^2]
N_g	number of years for power generator g to pay the overnight investment cost;
FOM_g	fixed O&M cost for power generator g; [k\$/MW]
KG_g	existing capacity of power generator g; [MW]
DF_g	derating factor of power generator g; [%]
$RM_{\hat{j}}$	reserve margin requirement for region $\hat{j}$; [%]
$PK_{\hat{j},t}^s$	peak level of hourly load in reserve margin region $\hat{j}$ during year t for scenario s. [MWh]

$$TC^{CE} = \sum_{s=1}^{S} p_s^{sc} \left\{ \sum_{g \in \mathbb{G}} \sum_{t=1}^{T} \delta^t \Big[\Big(\sum_{t' : t - N_g \le t' \le t} \frac{1}{N_g} \frac{1}{2} IC_{g,t'}^s x_{g,t'}^s \Big) + FOM_g k_{g,t}^s \Big] \right\}. \tag{15}$$

The decision variables have to satisfy the yearly cumulative constraints:

$$\begin{aligned} & k_{g,1}^s = KG_g, \\ & k_{g,t}^s = k_{g,t-1}^s + x_{g,t}^s, \quad t = 2, \ldots, T, \end{aligned} \quad \forall g \in \mathbb{G}, \quad s = 1, \ldots, S, \tag{16}$$

a reserve margin requirement constraint:

$$\begin{aligned} & \sum_{j \in \mathbb{J} : \mathcal{J}(j) = \hat{j}} \ \sum_{g \in \mathbb{G} : \mathcal{G}(g) = j} DF_g k_{g,t}^s \ge (1 + RM_{\hat{j}}) PK_{\hat{j},t}^s, \\ & \forall \hat{j} \in \hat{\mathbb{J}}, \quad t = 1, \ldots, T, \quad s = 1, \ldots, S, \end{aligned} \tag{17}$$

and the non-negativeness constraint:

$$x_{g,t}^s, k_{g,t}^s \ge 0, \quad \forall g \in \mathbb{G}, \quad t = 1, \ldots, T, \quad s = 1, \ldots, S. \tag{18}$$

Due to the presence of linkage between t and $t + 1$ in (16), the expanded capacity $x_{g,t}$ and the cumulative capacity $k_{g,t}$ are regarded as aggregated level decisions, as discussed in Section 2.1.

4.2 Sub-hourly Economic Dispatch of Generation and Transmission

We divided the set $\mathbb{G}$ into two groups, where $\mathbb{G}^{SR}$ denotes the set of power generators that cannot change their generation output levels within an hour, while $\mathbb{G}^{FR}$ denotes the set of power generators that can change their output levels quickly within an hour (the so-called fast ramping units).

In each year $t = 1, \dots, T$, the generation level $p_{g,h,t}$ of each slow-response generator $g \in \mathbb{G}^{SR}$ has to be decided for each hour $h = 1, \dots, H$, and the generation level $p_{g,m,t}$ of each fast-response generator $g \in \mathbb{G}^{FR}$ has to be decided for for each sub-hour $m = 1, \dots, M$. Consider a set of N nodes of the same scenario tree given in Section 4.1, and for each node $i = 1, \dots, N$, consider a set of R_i realizations of uncertainty $\boldsymbol{\eta}_i$. In this subsection, we describe the model of sub-hourly economic dispatch of generation and transmission. Table 6 summarizes all the indices, sets, and functions for the hourly/sub-hourly economic dispatch model; while the corresponding decision variables are listed in Table 7. The generation cost for slow-response generators is: $\sum_{g\in\mathbb{G}^{SR}} \sum_{h=1}^{H} VOM_g p_{g,h,i,r}$ for all $r = 1, \dots, R_i$ and $i = 1, \dots, N$, and the generation cost for fast response generators is: $\sum_{g\in\mathbb{G}^{FR}} \sum_{m=1}^{M} \frac{1}{M} VOM_g p_{g,m,i,r}$ for all $r = 1, \dots, R_i$ and $i = 1, \dots, N$. All parameters for sub-hourly economic dispatch of generation and transmission are listed in Table 8. There is also a penalty cost for the electricity outage if the generation supply could not meet the demand: $\sum_{j\in\mathbb{J}} \sum_{m=1}^{M} PENo_{j,m,i,r}$ for all $r = 1, \dots, R_i$ and $i = 1, \dots, N$. Then, the total cost of sub-hourly economic dispatch of generation and transmission is:

Table 6: Indices, Sets and Functions for Sub-hourly Economic Dispatch of Generation and Transmission

$\mathcal{T}$	set of nodes in a scenario tree, indexed i;
p_i^{nd}	probability of each node i;
R_i	number of realizations of uncertainty $\boldsymbol{\eta}_i$ at node i, indexed r;
p_r^{rl}	probability of each realization r;
$\mathbb{G}^{SR}$	set of slow-response generators;
$\mathbb{G}^{FR}$	set of fast-response generators;
$\mathcal{G}(g)$	function that maps a power generator g to a substation $j \in \mathbb{J}$;
$\mathbb{W}$	set of wind resources, indexed w;
$\mathcal{W}(w)$	function that maps a wind resource w to a substation $j \in \mathbb{J}$;
$\mathbb{D}$	set of demand nodes, indexed d;
$\mathcal{D}(d)$	function that maps a demand node d to a substation $j \in \mathbb{J}$;
$\mathbb{L}$	set of transmission lines connecting two substations, indexed l;
$\mathcal{L}_o(l)$	function that maps a transmission line l to an origin substation $j_o \in \mathbb{J}$;
$\mathcal{L}_d(l)$	function that maps a transmission line l to a destination substation $j_d \in \mathbb{J}$;
H	number of hours in a year, indexed h;
M	number of sub-hours in a year, indexed m;
$\mathcal{M}(m)$	function that maps a sub-hour m to an hour h.

$$TC^{ED} = \sum_{i=1}^{N} p_i^{nd} \left\{ \sum_{r=1}^{R_i} p_r^{rl} \delta^t \Big[\sum_{g\in\mathbb{G}^{SR}} \sum_{h=1}^{H} VOM_g p_{g,h,i,r} + \sum_{g\in\mathbb{G}^{FR}} \sum_{m=1}^{M} \frac{1}{M} VOM_g p_{g,m,i,r} + \sum_{j\in\mathbb{J}} \sum_{m=1}^{M} PENo_{j,m,i,r} \Big] \right\}. \tag{19}$$

The generation level of each power generator can not exceed its cumulative capacity:

$$\begin{aligned} &0 \le p_{g,h,i,r} \le k_{g,i,r}^{nr}, \quad \forall g \in \mathbb{G}^{SR}, \quad h = 1, \dots, H, \\ &0 \le p_{g,m,i,r} \le k_{g,i,r}^{nr}, \quad \forall g \in \mathbb{G}^{FR}, \quad m = 1, \dots, M, \end{aligned} \quad r = 1, \dots, R_i, \quad i = 1, \dots, N, \tag{20}$$

Table 7: Decision Variables for Sub-hourly Economic Dispatch of Generation and Transmission

$k^{nr}_{g,i,r}$	local copy of the cumulative capacity $k_{g,t(i)}$ of power generator g for realization r at node i;
$p_{g,h,i,r}$	generation level of slow-response generator $g \in \mathbb{G}^{SR}$ during hour h and year $t(i)$ for realization r at node i;[MW]
$p_{g,m,i,r}$	generation level of fast-response generator $g \in \mathbb{G}^{FR}$ during sub-hour m and year $t(i)$ for realization r at node i;[MW]
$f^{+}_{l,m,i,r}$	transmission flow on line l from substation j_o to j_d during sub-hour m and year $t(i)$ for realization r at node i;[MW]
$f^{-}_{l,m,i,r}$	transmission flow on line l from substation j_d to j_o during sub-hour m and year $t(i)$ for realization r at node i;[MW]
$\theta_{j,m,i,r}$	phase angle at substation j during sub-hour m and year $t(i)$ for realization r at node i;[rad]
$o_{j,m,i,r}$	outage at at substation j during sub-hour m and year $t(i)$ for realization r at node i.[MW]

Table 8: Parameters for Sub-hourly Economic Dispatch of Generation and Transmission

VOM_g	variable O&M cost for power generator g; [k$/MWh]
$a_{w,m,i,r}$	availability factor of wind resource $w \in \mathbb{W}$ during sub-hour m and year $t(i)$ for realization r at node i;
RU_g	ramp-up rate of cumulative capacity of power generator g;[%]
RD_g	ramp-down rate of cumulative capacity of power generator g;[%]
KW_w	capacity of wind resource w; [MW]
$D_{d,m,i,r}$	level of demand node d during sub-hour m and year $t(i)$ for realization r at node i;[MWh]
KL_l	capacity of transmission line l; [MW]
B_l	percentage of energy loss of transmission line l; [%]
Δ_l	susceptance of transmission line l, which equals the reciprocal of the reactance;
PEN	penalty cost of electricity outage. [k$/MW]

while still satisfies the ramping constraints:

$$\begin{aligned}
&p_{g,h,i,r} - p_{g,(h-1),i,r} \le RU_g k^{nr}_{g,i,r},\\
&p_{g,(h-1),i,r} - p_{g,h,i,r} \le RD_g k^{nr}_{g,i,r},\\
&\forall g \in \mathbb{G}^{SR}, \quad h = 2,\dots,H, \quad r = 1,\dots,R_i, \quad i = 1,\dots,N,
\end{aligned} \tag{21}$$

$$\begin{aligned}
&p_{g,m,i,r} - p_{g,(m-1),i,r} \le \frac{1}{M} RU_g k^{nr}_{g,i,r},\\
&p_{g,(m-1),i,r} - p_{g,m,i,r} \le \frac{1}{M} RD_g k^{nr}_{g,i,r},\\
&\forall g \in \mathbb{G}^{FR}, \quad m = 2,\dots,M, \quad r = 1,\dots,R_i, \quad i = 1,\dots,N.
\end{aligned} \tag{22}$$

The Kirchhoff's current law (KCL) has to be satisfied:

$$
\begin{aligned}
&\sum_{d\in\mathbb{D}:\mathcal{D}(d)=j} D_{d,m,i,r} + \sum_{l\in\mathbb{L}:\mathcal{L}_o(l)=j} \frac{1}{M} f^{+}_{l,m,i,r} + \sum_{l\in\mathbb{L}:\mathcal{L}_d(l)=j} \frac{1}{M} f^{-}_{l,m,i,r} \\
= &\sum_{g\in\mathbb{G}^{SR}:\mathcal{G}(g)=j} \frac{1}{M} p_{g,\mathcal{H}(m),i,r} + \sum_{g\in\mathbb{G}^{FR}:\mathcal{G}(g)=j} \frac{1}{M} p_{g,m,i,r} + \sum_{w\in\mathbb{W}:\mathcal{W}(w)=j} \frac{1}{M} a_{w,m,i,r} KW_w \\
+ &\sum_{l\in\mathbb{L}:\mathcal{L}_d(l)=j} \frac{1}{M}(1-B_l) f^{+}_{l,m,i,r} + \sum_{l\in\mathbb{L}:\mathcal{L}_o(l)=j} \frac{1}{M}(1-B_l) f^{-}_{l,m,i,r} + o_{j,m,i,r}, \\
&\forall j\in\mathbb{J}, \quad r=1,\dots,R_i, \quad i=1,\dots,N, \qquad (23)
\end{aligned}
$$

as well as the Kirchhoff's Voltage law (KVL):

$$
\begin{aligned}
&f^{+}_{l,m,i,r} - f^{-}_{l,m,i,r} = \Delta_l(\theta_{j_o,m,i,r} - \theta_{j_d,m,i,r}), \\
&\forall l\in\mathbb{L}, \quad m=1,\dots,M, \quad r=1,\dots,R_i, \quad i=1,\dots,N.
\end{aligned}
\qquad (24)
$$

The flow on each transmission line can not exceed its capacity:

$$
\begin{aligned}
0 \le f^{+}_{l,m,i,r} \le KL_l, \\
0 \le f^{-}_{l,m,i,r} \le KL_l,
\end{aligned}
\quad \forall l\in\mathbb{L}, \quad m=1,\dots,M, \quad r=1,\dots,R_i, \quad i=1,\dots,N, \qquad (25)
$$

and the phase angle and the electricity outage at each substation have to satisfy:

$$
\begin{aligned}
0 \le &\theta_{j,m,i,r} \le 2\pi, \\
&o_{j,m,i,r} \ge 0,
\end{aligned}
\quad \forall j\in\mathbb{J}, \quad m=1,\dots,M, \quad r=1,\dots,R_i, \quad i=1,\dots,N. \qquad (26)
$$

In spite of the operational coupling constraints, they are yearly independent, and hence are detailed level decisions.

4.3 A Co-optimization Model

Co-optimizing the long-term capacity expansion planning with the short-term sub-hourly economic dispatch of generation and transmission leads to the following multi-scale, multi-stage stochastic program:

$$
\begin{aligned}
\text{minimize} \quad & TC^{CE} + TC^{ED} \\
\text{subject to} \quad & (16) - (18), \quad (20) - (26) \\
& \sum_{s=1}^{S} K_s \mathbf{k}^s_g + \sum_{i=1}^{N}\sum_{r=1}^{R_i} K^{nr}_{i,r} k^{nr}_{g,i,r} = \mathbf{0}, \quad \forall g\in\mathbb{G},
\end{aligned}
\qquad (27)
$$

where $\mathbf{k}^s_g := (k^s_{g,1} \cdots k^s_{g,T})^T$ for all $s=1,\dots,S$. The last equality constraint is the extended nonanticipativity constraint we proposed in Section 2.3. Problem (27) is of the same form as (6), and hence can be solved by the simplified N-block PCPM algorithm, proposed in Algorithm 1. Moreover, we observe that the constraint (17) can be removed for the decision variables $k^s_{g,t}$'s and added to the node decision variables $k^{nr}_{g,i,r}$'s as follows:

$$
\begin{aligned}
&\sum_{j\in\mathbb{J}:\mathcal{J}(j)=\hat{j}} \; \sum_{g\in\mathbb{G}:\mathcal{G}(g)=j} DF_g k^{nr}_{g,i,r} \ge (1+RM_{\hat{j}}) PK_{\hat{j},i}, \\
&\forall \hat{j}\in\hat{\mathbb{J}}, \quad r=1,\dots,R_i, \quad i=1,\dots,N.
\end{aligned}
\qquad (28)
$$

Minimizing TC^{CE} subject to (16) and (18) can then be decomposed by both power generator $g\in\mathbb{G}$ and scenario $s=1,\dots,S$. Minimizing TC^{ED} subject to (20) to (26), as well as (28), can then be decomposed by both realization $r=1,\dots,R_i$ and node $i=1,\dots,N$.

5 Numerical Experiments

5.1 Data

We consider a planning horizon of $T = 7$ years, as well as a discount factor $\delta = \frac{1}{1.02}$. For each of 12 months in a year, we consider 1 representative day, which can be further divided in 24 hours or 288 5-minutes. Hence, the total number of hours in a year is: $H = 12 \times 24 = 288$, and the total number of sub-hours (5-minutes) in a year is $M = 12 \times 288 = 3456$.

We focus on a single reserve margin region with $RM = 0.15$. In this region, we consider: 17 substations, 11 generators, 1 wind resource, 14 demand nodes and 25 transmission lines, which form a double-loop network plotted in Fig 5. To generate the network, we use a synthetic data set of Texas network[1] for reference.

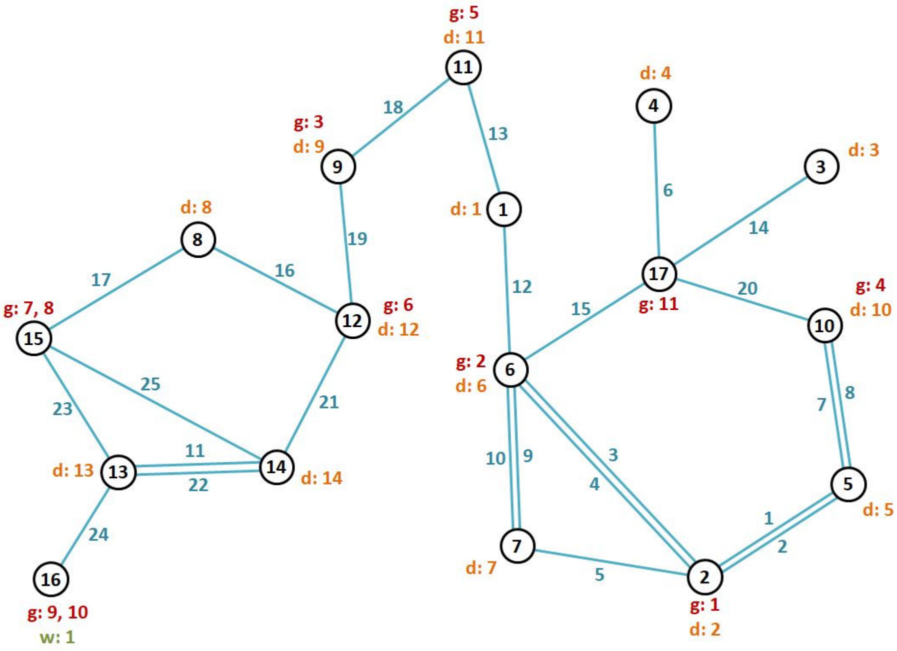

Figure 5: A double-loop Network in a Reserve Margin Region.

First, we partially select 17 substations from the north central area, as well as 11 power generators, 14 demand nodes and 25 transmission lines, from the original data set. Then, we manually assign a technology type to each power generator, with parameters set based on the historical data of EIA Annual Energy Outlook (EIA AEO)[2]. The detailed values of all parameters for the power generators can be found in Table 9. A single wind resource is manually added to Substation 16 with a capacity of $KW = 8.0MW$. To each demand node d in the network, we assign a load fraction LF_d, denoting the ratio of the hourly load at node d to the total hourly load of the whole north central area of Texas, listed in Table 10. All detailed values of the parameters for the transmission lines can be found in Table 11.

5.2 Scenario Tree Generation

The way of generating a scenario tree, presented in this subsection, is only to illustrate the numerical performance of the proposed algorithm, which will be shown in the next subsection. For more thorough studies on capacity expansion planning for a particular region, we will be using a similar method as in [3], where the

[1] The data can be found at https://electricgrids.engr.tamu.edu/.

[2] The data can be found at https://www.eia.gov/outlooks/aeo/.

Table 9: Data of Generation Sectors.

g	**Resp.**	j	**Type**	KG_g	N_g	FOM_g	VOM_g	RU_g	RD_g	DF_g
1	fast	2	Conventional CT	20.0	30	7.34	0.01545	100%	100%	1.0
2	fast	6	Concentional CC	40.0	30	13.17	0.0036	33.33%	33.33%	
3	fast	9	Advanced CC	30.0	30	15.37	0.00327	50%	50%	
4	fast	10	Advanced CT	10.0	30	7.04	0.01037	100%	100%	
5	fast	11	Advanced CC	20.0	30	15.37	0.00327	50%	50%	
6	fast	12	Advanced CT	10.0	30	7.04	0.01037	100%	100%	
7	slow	15	Single Advanced PC	40.0	40	37.8	0.00447	0	25%	
8	slow	15	Dual Advanced PC	60.0	40	31.18	0.00447	0	25%	
9	slow	16	Single Advanced PC	40.0	40	37.8	0.00447	0	25%	
10	slow	16	Dual Advanced PC	60.0	40	31.18	0.00447	0	25%	
11	slow	17	Nuclear	180.0	40	93.28	0.00214	0	25%	

Table 10: Data of Demand Nodes.

d	j	LF_d	d	j	LF_d
1	1	0.00341506840912519	8	8	0.00154893798556277
2	2	0.0101280540133409	9	9	0.00261269311300064
3	3	0.003343230400519	10	10	0.00354824502507975
4	4	0.00296193635483998	11	11	0.000419423450246929
5	6	0.000911237509166254	12	12	0.000151412418139208
6	6	0.0000204462024494551	13	13	0.0000569178068187533
7	7	0.00826579179024186	14	14	0.000143123417146185
		$\sum_{d=1}^{14} LF_d$		0.0375265178956769	

Table 11: Data of Transmission Lines.

l	j_o	j_d	B_l	Δ_l	KL_l	l	j_o	j_d	B_l	Δ_l	KL_l
1	2	5	0.287%	126.524	150.0	14	3	17	13.934%	103.7115	1327.0
2	2	5	0.287%	126.524	150.0	15	6	17	31.84%	43.83385	1327.0
3	2	6	0.833%	43.62298	150.0	16	8	12	28.678%	64.59031	1327.0
4	2	6	0.833%	43.62298	150.0	17	8	15	30.816%	59.91346	1327.0
5	2	7	14.845%	201.1213	310.0	18	9	11	24.93%	56.07047	1327.0
6	4	17	0.683%	50.40032	150.0	19	9	12	45.045%	40.93052	1327.0
7	5	10	0.29%	125.1586	150.0	20	10	17	31.441%	44.39812	1327.0
8	5	10	0.29%	125.1586	150.0	21	12	14	58.379%	24.80828	1327.0
9	6	7	0.416%	82.76678	150.0	22	13	14	38.068%	36.6839	1327.0
10	6	7	0.416%	82.76678	150.0	23	13	15	12.077%	115.5839	1327.0
11	13	14	3.753%	6.12558	221.0	24	13	16	10.063%	119.5711	1494.0
12	1	6	22.561%	64.15568	1327.0	25	14	15	44.721%	31.2298	1327.0
13	1	11	9.567%	193.4949	1327.0						

stochastic process is modeled as a geometric browning motion with the parameters verified by the historical data. It will be one of our future research tasks.

We consider a 7-level binary tree, where each node at level $t = 1, \dots, 6$ has only 2 child nodes, so there are $S = 64$ scenarios and $N = 127$ nodes in total. With each node $i = 1, \dots, 127$, we associate the data vectors: $\vec{IC}_i$, PK_i^{NC}, $\vec{D}_i^{NC,WKD}$ and $\vec{D}_i^{NC,WKE}$, where each component of $\vec{IC}_i := (IC_{1,i} \cdots IC_{11,i})^T$ denotes the quadratic coefficient $IC_{g,i}$ for each generation sector g at node i, PK_i^{NC} denotes the peak level of hourly load in the north central area at node i, each component of $\vec{D}_i^{NC,WKD} := (D_{1,i}^{NC,WKD} \cdots D_{12,i}^{NC,WKD})^T$ denotes the average weekday load $D_{mon,i}^{NC,WKD}$ of month $mon = 1, \dots, 12$ in the north central area at node i, and each component of $\vec{D}_i^{NC,WKE} := (D_{1,i}^{NC,WKE} \cdots D_{12,i}^{NC,WKE})^T$ denotes the average weekend load $D_{mon,i}^{NC,WKD}$ of month mon in the north central area at node i.

The probability of creating each child node i_c from its ancestor node i at level $t = 1, \dots, T-1$, denoted by $p_{i,ic}$, has 2 different values, listed in Table 12. We consider the year of 2019 as the beginning of the planning horizon. At the root node 1, with probability $p_1 = 1.0$, the values of $\vec{IC}_1$ is obtained from the historical data of EIA AEO, and the value of PK_1^{NC} is obtained by ERCOT hourly load data from the year of 2019, as well as the values of $\vec{D}_1^{NC,WKD}$ and $\vec{D}_1^{NC,WKE}$. Each child node i_c is created with a probability $p_{i_c} = p_i \times p_{i,i_c}$. The values of the data associated with each child node i_c is calculated using the values associated with its ancestor node times a ratio, also listed in Table 12. The ratio $\frac{IC_{g,i_c}}{IC_{g,i}}$ at each child node i_c is manually assigned. The ratio $\frac{PK_i^{NC}}{PK_{i_c}^{NC}}$ at each child node i_c is generated from the ERCOT hourly load data for the year of 2002 to 2019, as well as the ratios $\frac{\vec{D}_{i_c}^{NC,WKD}}{\vec{D}_i^{NC,WKD}}$ and $\frac{\vec{D}_{i_c}^{NC,WKE}}{\vec{D}_i^{NC,WKE}}$ at each child node i_c. Then, a 7-level binary tree can be generated. The values of $IC_{g,t}^s$ can be found at the corresponding node, and $PK_i = \left(\sum_{d=1}^{14} LF_d\right) \times PK_i^{NC}$.

In our model, we also consider an uncertainty for the sub-hourly load and the wind availability factor. At each node $i = 1, \dots, 127$, we consider 10 realizations with a equal probability $p_r^{rl} = 0.1$. For each month $mon = 1, \dots, 12$, two data pools of 5-minute load portions to a daily load are made for weekdays and weekend days respectively, using the historical data of ERCOT. For each realization r, a sequence of 12 representative days in each month is sampled. For each representative day in each month $mon = 1, \dots, 12$, a sequence of total 288 5-minutes load portions in a day is drawn from a data pool, depending on the day type. Then, $D_{d,m,i,r}$ is calculated by multiplying $D_{mon,i}^{NC,WKD}$ or $D_{mon,i}^{NC,WKE}$, whose month mon contains the sub-hour m, with the portion in sub-hour m and the load fraction LF_d. Similarly, for each representative day in each month $mon = 1, \dots, 12$, a sequence of 288 average wind availability factor during 5 minutes in a day is drawn from a data pool corresponding to month mon, made by the historical data of WIND Toolkit from NREL[3]. The whole sequence of 12 months determines the values of $a_{m,i,r}$.

5.3 Numerical Results

We compare the performance of the simplified PCPM algorithm with that of the ADMM algorithm and the PHA algorithm, shown in Table 13. For both simplified PCPM and ADMM, τ measures the average residual of the extended nonanticipativity constraints; that is:

$$\tau = \frac{1}{\sqrt{n_{sc}^Y + n_{nr}^Y}} \left\| \sum_{s=1}^{S} K_s \mathbf{y}^{s,k} + \sum_{i=1}^{N} \sum_{r=1}^{R_i} K_{i,r}^{nr} \mathbf{y}_{i,r}^{nr,k} \right\|_2.$$

For PHA, τ measures the average residual of the original nonanticipativity constraints; that is: $\tau = \frac{1}{\sqrt{n_{sc}}} \left\| \sum_{s=1}^{S} K_s \mathbf{x}^{s,k} \right\|_2$.

Compared with PHA, simplified PCPM, as well as ADMM, decomposes the large-scale problem by both scenario and node-realization, while PHA only decomposes the problem by scenario. Using the hybrid decomposition method, the original problem is decomposed into 1334 sub-problems, whose largest size of

[3] The data can be found at https://www.nrel.gov/grid/wind-toolkit.html.

Table 12: Probabilities and Ratios for Generating the Binary Scenario Tree.

	Decreasing		**Increasing**			
	p_{i,i_c}				p_1	
	0.411765		0.588235		1.0	
g	$\frac{IC_{g,i_c}}{IC_{g,i}}$				$IC_{g,1}$	
1	0.999		1.07		4.87	
2	0.999		1.06		1.48	
3	0.999		1.04		2.56	
4	0.999		1.05		3.22	
5	0.999		1.04		2.56	
6	0.999		1.05		3.22	
7	0.999		1.08		4.99	
8	0.999		1.09		2.26	
9	0.999		1.08		4.99	
10	0.999		1.09		2.26	
11	1.0		1.0		2.48	
	$\frac{PK^{NC}_{i_c}}{PK^{NC}_{i}}$				PK^{NC}_1	
	0.962635		1.047867		25493.791364	
mon	$\frac{D^{NC,WKD}_{mon,i_c}}{D^{NC,WKD}_{mon,i}}$	$\frac{D^{NC,WKE}_{mon,i_c}}{D^{NC,WKE}_{mon,i}}$	$\frac{D^{NC,WKD}_{mon,i_c}}{D^{NC,WKD}_{mon,i}}$	$\frac{D^{NC,WKE}_{mon,i_c}}{D^{NC,WKE}_{mon,i}}$	$D^{NC,WKD}_{mon,1}$	$D^{NC,WKE}_{mon,1}$
1	0.939811	0.922987	1.079484	1.095487	328060.997010	314257.280135
2	0.922397	0.933038	1.087484	1.086450	312898.959607	301457.547991
3	0.976826	0.964011	1.042382	1.049035	286408.897311	273070.060138
4	0.965699	0.946616	1.037831	1.055578	270477.852282	250265.200286
5	0.948244	0.928000	1.059238	1.075360	313919.847511	285299.536478
6	0.940542	0.937028	1.058858	1.073340	360777.718191	352160.898949
7	0.941563	0.933001	1.065955	1.072776	407644.239217	391658.771474
8	0.946417	0.936104	1.064896	1.072610	446160.028895	412306.627903
9	0.942981	0.928792	1.074786	1.092140	409638.432070	394556.334726
10	0.970548	0.953333	1.047375	1.062817	305789.782162	283395.415091
11	0.974836	0.966007	1.045072	1.052684	295388.955169	268610.114623
12	0.951374	0.936434	1.058845	1.073102	308500.818307	276917.790869

the decision variable is approximately 3×10^5, with a largest number of constraints being around 2×10^5. However, using only scenario decomposition, the problem can only be decomposed into 64 sub-problems, causing a fact that both the size of the decision variable and the number of constraints are almost 70 times of that in the hybrid decomposition. A much larger-sized sub-problem not only requires more amount of memory but also takes more time to be solved in each iteration.

Within a 9-hour usage of computing resources on a multi-node computer cluster, PHA is implemented on 64 processors, each of which corresponds to a computing unit that solves a sub-problem. The 64 processors are mapped to 64 nodes with 1 processor per node. The memory usage on each node is about 20 GB. The algorithm terminates with a much larger average residual, as well as much higher total cost and outage penalty, compared with simplified PCPM and PHA.

Table 13: Numerical Results for Sub-hourly Modeling on a multi-node computer cluster.

Algorithm	**Simplified PCPM**	**ADMM**	**PHA**
decomposed by	scenario and node-realization	scenario and node-realization	scenario only
total number of decomposed sub-problems	1334	1334	64
largest size of decision variable in each decomposed sub-problem	$312,491$	$312,491$	$21,873,754$
largest number of constraints in each decomposed sub-problem	$211,417$	$211,417$	$14,799,204$
total number of processors	1334	1334	64
total number of nodes (maximum 20 cores per node)	67	67	64
maximum memory per node	6.8 GB	6.8 GB	20.4 GB
total elapsed wall-clock time	6.92 h	6.98 h	8.57 h
total number of iterations	1196	1228	6
τ	0.0001	0.0002	1.2150
total cost	$233,998.562$ (k$)	$234,497.601$ (k$)	$235,795.482$ (k$)
outage penalty	0.025424 (k$)	0.030819 (k$)	7.961365 (k$)

Both simplified PCPM and ADMM are implemented on 1334 processors, each of which solves a decomposed sub-problem. The 1334 processors are mapped to 67 nodes with maximum 20 cores per node. Compared with PHA, both two algorithms use much less memory per node, which demonstrates the strengths of using hybrid decomposition. Additionally, compared with ADMM, simplified PCPM converges with fewer number of iterations and less elapsed wall-clock time, due to the benefits of exploiting the technique of orthogonal projection. The algorithm also terminates with a smaller average residual, as well as lower total cost and outage penalty.

6 Conclusion and Future Works

In this paper, we apply the N-block PCPM algorithm to solve multi-scale multi-stage stochastic programs, with the application to electricity capacity expansion models. Numerical results show that the proposed simplified N-block PCPM algorithm, along with the hybrid decomposition method, exhibits much better scalability for solving the resulting deterministic, large-scale block-separable optimization problem, when compared with the ADMM algorithm and the PHA algorithm. The superiority of the algorithm's scalability is attributed to the two key features of the algorithm design: first, the proposed hybrid scenario-node-realization decomposition method with extended nonanticipativity constraints can decompose the original problem under various uncertainties of different temporal scales; second, when applying the N-block PCPM algorithm to solve the resulting deterministic, large-scale N-block convex optimization problem, the technique of orthogonal projection we exploit greatly simplifies the iteration steps and reduce the communication overhead among all computing units, which also contributes to the efficiency of the algorithm.

Numerical experiments with better ways of scenario generation will be conducted in the future. Retirement of generators, as well as storage of electricity, will be considered in future models. The number of substations will also be increased, when more computing resources become available.

References

[1] Gong Chen and Marc Teboulle. A proximal-based decomposition method for convex minimization problems. *Mathematical Programming*, 64(1-3):81–101, 1994.

[2] Kjetil Høyland and Stein W. Wallace. Generating scenario trees for multistage decision problems. *Management Science*, 47(2):295–307, 2001.

[3] Shan Jin, Sarah M Ryan, Jean-Paul Watson, and David L Woodruff. Modeling and solving a large-scale generation expansion planning problem under uncertainty. *Energy Systems*, 2(3-4):209–242, 2011.

[4] Teemu Pennanen. Epi-convergent discretizations of multistage stochastic programs via integration quadratures. *Mathematical Programming*, 116(1-2):461–479, 2009.

[5] G. Ch. Pflug. Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical Programming*, 89(2):251–271, 2001.

[6] R. Tyrrell Rockafellar and Roger J.-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16(1):119–147, 1991.

[7] Werner Römisch. Scenario generation. *Wiley Encyclopedia of Operations Research and Management Science*, 2010.

[8] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2014.