Multistage Stochastic Unit Commitment Using Stochastic Dual Dynamic Integer Programming

Jikai Zou, Shabbir Ahmed, *Senior Member*, and Andy Sun, *Senior Member*

***Abstract*—Unit commitment (UC) is a key operational problem in power systems for the optimal schedule of daily generation commitment. Incorporating uncertainty in this already difficult mixed-integer optimization problem introduces significant computational challenges. Most existing stochastic UC models consider either a two-stage decision structure, where the commitment schedule for the entire planning horizon is decided before the uncertainty is realized, or a multistage stochastic programming model with relatively small scenario trees to ensure tractability. We propose a new type of decomposition algorithm, based on the recently proposed framework of Stochastic Dual Dynamic Integer Programming (SDDiP), to solve the multistage stochastic unit commitment (MSUC) problem. We propose a variety of computational enhancements to SDDiP, and conduct systematic and extensive computational experiments to demonstrate that the proposed method is able to handle elaborate stochastic processes and can solve MSUCs with a huge number of scenarios that are impossible to handle by existing methods.**

***Index Terms*—Unit commitment, multistage stochastic integer programming, stochastic dual dynamic integer programming**

NOMENCLATURE

Indices

n, m	Node in the scenario tree
i	Generation unit
b	Load bus
ℓ	Transmission line
t	Decision stage
$t(n)$	Decision stage of node n

Sets

$\mathcal{T}$	Scenario tree
$\mathcal{B}$	Set of all buses
$\mathcal{G}$	Set of generation units
$\mathcal{G}_b$	Set of generation units at bus b
$\mathcal{D}$	Set of demand bus
$\mathcal{L}$	Set of all transmission lines
$\mathcal{P}(n,t)$	Path from the t-th ancestor node of n to n

Parameters

p_n	Probability associated with node n
$\overline{S}_i$, $\underline{S}_i$	Start-up/shut-down cost of generation unit i
C_p	Penalty cost
D_{nb}	Load demand of bus b at node n
F_l	Maximum flow capacity of line ℓ
K_l	Shift vectors of transmission line ℓ
$\overline{P}_i$, $\underline{P}_i$	Max/min output of generation unit i
R_t	Reserve requirement in period t
$\overline{\Delta}_i$, $\underline{\Delta}_i$	Regular ramping rate for generation unit i
Δ^{SU}, Δ^{SD}	Start-up/shut-down ramping rate for unit i
UT_i, DT_i	Minimum up/down time for generation unit i

Variables

x_{ni}	State of unit i at node n, equals 1 if on, 0 otherwise
y_{ni}	Generation by unit i at node n
u_{ni}	Start-up decision for unit i, equals 1 if it is turned on at node n, 0 otherwise
v_{ni}	Shut-down decision for unit i, equals 1 if it is turned off at node n, 0 otherwise
r_{ni}	Reserved spinning capacity from unit i at node n
δ_n^+	Total unsatisfied demand at node n
δ_n^-	Total over-generation at node n

J. Zou, S. Ahmed, A. Sun are with the Department of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA e-mail: jikai.zou@gatech.edu, shabbir.ahmed, andy.sun@isye.gatech.edu. This research has been supported in part by the National Science Foundation Grant 1633196.

I. INTRODUCTION

A. Motivation

UNIT commitment (UC) is one of the key operational problems in power systems. It is used by system operators to decide a commitment schedule of generation units for the next day or week, under which the forecast demand can be met in the most cost efficient way. Besides satisfying electricity load, the commitment decisions also need to satisfy various physical constraints, such as generation capacity, minimum up and down time, ramping limit, as well as the flow limit of transmission lines.

In recent years, an increasing penetration of renewable energy has cast another layer of complexity to the UC problem. Due to the intermittent and stochastic nature renewable energy, power system operators need to model uncertainty and to solve the resulting large-scale computation problems. Stochastic optimization has been utilized in UC problems to achieve this goal. There are two types of stochastic models, namely *two-stage* stochastic UC model and *multistage* stochastic UC model, that have been explored. In a two-stage stochastic UC model, the UC decision is the first-stage decision, which is determined in the day before the uncertainty is realized and thus is not adaptive to specific uncertainty realizations. In contrast, a multistage stochastic UC (MSUC) model handles uncertainty dynamically in the sense that the UC decision is a function of the realization of load and renewable generation, i.e. the UC decision adapts to uncertainty realizations.

There exists an extensive literature on two-stage and multistage stochastic UC models. We refer the reader to the comprehensive surveys [1; 2; 3] for the progress in the two-stage setting, and summarize related work in multistage stochastic UC in the next section.

B. Related Work on Multistage Stochastic UC

The benefit of the multistage approach in terms of increasing system flexibility and reducing operation costs compared to the deterministic or two-stage approaches has been successfully demonstrated in various previous works [3]. In a recent work [4] the authors provide a theoretical analysis on the value of multistage stochastic optimization in UC. They also provide a numerical study on a small 10 generator instance from the literature and show that the gap between the objective of the two-stage and multistage model can be up to 3.5%. However this advantage comes at the expense of significant computational cost.

A major line of research on MSUC has been the development of advanced decomposition algorithms. Two different approaches have been proposed, namely *unit decomposition* and *scenario decomposition*. In unit decomposition, constraints that couple generation units, such as load balancing, transmission, and spinning reserve constraints, are relaxed so that each subproblem corresponds to a single generation unit. Such a decomposition scheme was first studied in [5] and has been extended by incorporating electricity contracts and spot-market prices [6]. Proximal bundle method (cf. [7]) is used in solving the Lagrangian dual problem [8; 9]. A Dantzig-Wolfe decomposition approach is studied in [10], where the single-unit subproblems are solved by dynamic programming and their schedules are added back to the restricted master problem.

Alternatively, the scenario decomposition approach attempts to relax the coupling constraints among scenarios, usually referred to as non-anticipativity constraints. The ensuing subproblems then correspond to single scenarios. Different methods have been applied to solve the relaxed problem, such as the progressive hedging algorithm [11] and Dantzig-Wolfe decomposition [12; 13]. Besides the development of decomposition techniques, there have been efforts on strengthening the formulation using effective cutting planes. A majority of these work focus on the two-stage setting (e.g., [14; 15; 16], etc.). Recently, cutting planes have been studied for multistage models [17; 18].

The previously discussed solution methods are usually applicable only to relatively small scenario trees, and can quickly run into scalability issues on practical problems with large scenarios trees. To deal with large scenario trees, Pereira and Pinto [19] propose a popular, sampling-based variant of nested Benders decomposition, known as Stochastic Dual Dynamic Programming (SDDP) to solve multistage stochastic linear programs. It has been widely applied to multistage stochastic hydro-thermal scheduling problems (see e.g. [20; 21; 22; 23; 24; 25].) Existing attempts to extend SDDP to multistage stochastic mixed-integer programs have focused on exploiting convex relaxations of the cost-to-go function (see e.g. [26] for approximating the bilinear terms by the McCormick envelopes and [27] for using Lagrangian relaxation to smoothen the nonconvex shape of the cost-to-go functions).

To overcome the intrinsic limitation of Benders-type decomposition for multistage stochastic integer programs, a new algorithm called Stochastic Dual Dynamic integer Programming (SDDiP) is developed in [28], which guarantees to find the exact optimal solution of a multistage stochastic mixed integer program with binary state variables and mixed-integer recourse variables. This is accomplished by a new family of valid cuts, termed Lagrangian cuts, which are able to achieve strong duality for mixed integer programs. SDDiP is also sampling-based algorithm as SDDP, and exhibits promising scalability on solving large-scale scenario trees.

C. Contributions

In this paper, we adapt the SDDiP algorithm to solve large-scale MSUC problems under load and renewable generation uncertainty. The key contributions of the paper are summarized below.

1) We propose a dynamic programming based reformulation of MSUC to adapt the SDDiP algorithm for a stage-wise decomposition of the problem. To the best of our knowledge, this is the first attempt to tackle the MSUC problem using a stage-wise decomposition framework. The proposed algorithm guarantees to return a multistage unit commitment policy that the operators can use in the *real time* commitment and dispatch to increase system flexibility.
2) We propose several algorithmic development to significantly reduce computation time of the SDDiP algorithm, including (i) using a stabilized cutting plane algorithm, the so-called level method, to compute the Lagrangian cuts, (ii) a "hybrid" mixed-integer and linear modeling approach with the notion of "breakstage" to reduce the effective size of the scenario tree, and (iii) a parallel implementation that significantly speeds up the backward sweep of the proposed algorithm.
3) Extensive computational experiments are conducted on the IEEE 14-bus and 118-bus systems. We study the effectiveness of various valid inequalities for solving the MSUC problem and the impact of breakstage. Our experiments show that the proposed method can handle MSUCs with a huge number of scenarios that existing algorithms cannot handle.

The remainder of the paper is organized as follows. In Section II, we describe the SDDiP algorithm and different cut families. In Section III, we present the mathematical formulation for MSUC. In Section IV, we describe various computational enhancements of SDDiP for solving MSUC. Sections V-VI discuss experiment settings and detailed computational results. Finally, we provide some concluding remarks in Section VII.

II. A Primer on SDDiP

A. Multistage Stochastic Integer Program

We start with a scenario tree formulation of a multistage stochastic integer problem (MSIP).

$$\min_{x_n, y_n, z_n} \quad \sum_{n \in \mathcal{T}} p_n g_n(x_n, y_n) \tag{1a}$$

$$\text{s.t.} \quad \forall n \in \mathcal{T}$$

$$(z_n, x_n, y_n) \in X_n \tag{1b}$$

$$z_n = x_{a(n)}, z_n \in [0,1]^d \tag{1c}$$
$$x_n \in \{0,1\}^d, \tag{1d}$$

where x_n is the linking variable, also called state variable, of node n in the scenario tree $\mathcal{T}$, because node n's decision x_n is linked to its ancestor node $a(n)$'s state $x_{a(n)}$ through (1b) and (1c); y_n is a local variable that only appears in node n's problem and (1c) is a redundant constraint that makes a local copy z_n of the ancestor's state $x_{a(n)}$. This redundant constraint turns out to be crucial for the success of SDDiP as we will discuss below.

This formulation (1) is called *multistage* because the state and local decisions x_n, y_n are associated with *each* node in the scenario tree, therefore, can adapt to each specific nodal realization of uncertainty. In terms of the UC problem, such a formulation allows the UC decision at each hour to adapt to specific uncertainty history up to that hour. Hence, by solving (1), we get a *policy* or a *contingency plan* on how to commit and dispatch generators for each possible realization of scenarios. Such a policy is extremely flexible and comprehensive. An operator can tailor this UC policy according to specific operating environment.

B. Dynamic Programming Reformulation

Now we can write down the dynamic programming (DP) reformulation for the multistage problem (1) as follows: For root node $n = 1$ of the scenario tree,

$$(P_1) \quad \min \quad g_1(x_1, y_1) + \sum_{m \in \mathcal{C}(1)} q_{1m} Q_m(x_1)$$
$$\text{s.t.} \quad \text{(1b)} - \text{(1d)} \text{ (for } n = 1)$$

and for each node $n \in \mathcal{T} \setminus \{1\}$,

$$(P_n) \quad Q_n(x_{a(n)}) := \min \; g_n(x_n, y_n) + \sum_{m \in \mathcal{C}(n)} q_{nm} Q_m(x_n)$$
$$\text{s.t. (1b)} - \text{(1d)},$$

where q_{nm} is the conditional probability from n to its children node m. We will refer to $Q_n(\cdot)$ as the optimal value function (of $x_{a(n)}$) at node n and denote the function $\mathcal{Q}_n(\cdot) := \sum_{m \in \mathcal{C}(n)} q_{nm} Q_m(\cdot)$ as the expected cost-to-go (ECTG) function at node n.

C. SDDiP Algorithm

The SDDiP algorithm assumes the scenario tree to be stage-wise independent, i.e., for any two nodes n and n' in stage t, the set of their children nodes $\mathcal{C}(n)$ and $\mathcal{C}(n')$ have the same conditional probability distribution. Stage-wise dependency can be modeled as stage-wise independence by properly extending the state space. Once we have stage-wise independence, the ECTG functions depend only on the stage rather than the nodes, i.e., $\mathcal{Q}_n(\cdot) \equiv \mathcal{Q}_t(\cdot)$ for all n in stage t. Each iteration of SDDiP starts with sampling a subset of scenarios from the scenario tree (Line 3 in Algorithm 1.)

1) Forward Step: In the forward step, SDDiP algorithm proceeds stage-wise from $t = 1$ to T on the sampled scenarios and solves DP recursion on an approximate convex piecewise linear ECTG function at each stage. In particular, the stage problem, $P_t^i(x_{t-1}^i, \psi_t^i, \xi_t^k)$, in the i-th iteration is of the following form:

$$\underline{Q}_t^i(x_{t-1}^i, \psi_t^i, \xi_t^k) := \min_{x_t, y_t, z_t} \; g_t(x_t, y_t, \xi_t^k) + \psi_t^i(x_t) \tag{2a}$$
$$\text{s.t.} \; (x_t, y_t, z_t) \in X_t(\xi_t^k) \tag{2b}$$
$$z_t = x_{t-1}^i \tag{2c}$$
$$x_t \in \{0,1\}^d, \; z_t \in [0,1]^d, \tag{2d}$$

where ξ_t^k is the k uncertainty realization at stage k and the approximate ECTG function $\psi_t^i(\cdot)$ is defined as:

$$\psi_t^i(x_t) := \min \big\{\theta_t : \quad \theta_t \geq L_t, \tag{3a}$$
$$\theta_t \geq \frac{1}{N_{t+1}} \sum_{j=1}^{N_{t+1}} (v_{t+1}^{\ell j} + (\pi_{t+1}^{\ell j})^\top x_t), \; \forall \ell \leq i-1 \big\}. \tag{3b}$$

The function $\psi_t^i(x_t)$ is the current convex lower-approximation of the true ECTG function $\mathcal{Q}_t(x_t)$. Once a forward iteration is completed, we have obtained a feasible solution $\{(x_t^i, y_t^i)\}_{t=1}^T$ for the corresponding scenario.

2) Backward Step:: The backward step starts from the last stage T. Given the solution x_{T-1}^i from iteration i and a particular uncertainty realization $\xi_T^j (1 \leq j \leq N_T)$, let R_T^{ij} be a relaxation of the forward problem $P_T^i(x_{T-1}^i, \psi_T^{i+1}, \xi_T^j)$. Note that $\psi_T^i \equiv 0$ for all $i \geq 0$. Solving R_T^{ij} for each j produces a linear inequality defined by (v_T^{ij}, π_T^{ij}) and it is valid for the value function $Q_T(x_{T-1}, \xi_T^j)$. Then the inequalities are aggregated to obtain one of form (3b), which is valid for ECTG function $\mathcal{Q}_{T-1}(x_{T-1})$. The lower approximation of the ECTG function is updated from $\psi_{T-1}^i(\cdot)$ to $\psi_{T-1}^{i+1}(\cdot)$. The backward step then proceeds to stage $T-1$. When the first stage is completed, since we have solved a lower approximation of the original problem, the optimal value of the first stage problem is a valid lower bound of the original problem. A complete description of the SDDiP algorithm is given in Algorithm 1.

A stopping criterion commonly used for SDDP-type algorithms is to terminate when the gap $UB - LB$ in Algorithm 1 is small. Since UB is a statistical upper bound, some precaution is needed for this criterion. See [29] for some discussion. We use another simple stopping criterion to terminate the algorithm once the lower bound stabilizes. See [28] for more discussions.

D. Cut Families in Backward Step

Different cutting plane methods can be used in R_t^{ij} in the backward step. In this paper, we investigate three types: standard Benders cuts, a type of Lagrangian cuts obtained from a particular reformulation, and strengthened Benders cuts, which are a byproduct of the Lagrangian cuts.

1) Benders cut: Benders cuts are widely used in stochastic LP, but are also used in stochastic IP, providing a linear relaxation to the IP problem. The main drawback is that Benders cut are in general not tight for IP. Therefore, convergence of SDDiP with Benders cuts is not guaranteed.

Algorithm 1 :: Stochastic Dual Dynamic Integer Programming

Initialize: $LB \leftarrow -\infty$, $UB \leftarrow +\infty$, $i \leftarrow 1$, and an initial under-approximation $\{\psi_t^1(\cdot)\}_{t=1,\dots,T}$
while some stopping criterion is not satisfied **do**
  Sample M scenarios $\Omega^i = \{\xi_1^k, \dots, \xi_T^k\}_{k=1\dots,M}$
  /* Forward step */
  for $k = 1, \dots, M$ **do**
    for $t = 1, \dots, T$ **do**
      solve forward problem $P_t^i(x_{t-1}^{ik}, \psi_t^i, \xi_t^k)$
      collect solution $(x_t^{ik}, y_t^{ik}, z_t^{ik}, \theta_t^{ik} = \psi_t^i(x_t^{ik}))$
    end for
    $u^k \leftarrow \sum_{t=1,\dots,T} g_t(x_t^{ik}, y_t^{ik}, \xi_t^k)$
  end for
  /* Statistical upper bound update */
  $\hat{\mu} \leftarrow \frac{1}{M}\sum_{k=1}^{M} u^k$ and $\hat{\sigma}^2 \leftarrow \frac{1}{M-1}\sum_{k=1}^{M}(u^k - \hat{\mu})^2$
  $UB \leftarrow \hat{\mu} + z_{\alpha/2}\frac{\hat{\sigma}}{\sqrt{M}}$
  /* Backward step */
  for $t = T, \dots, 2$ **do**
    for $k = 1, \dots, M$ **do**
      for $j = 1, \dots, N_t$ **do**
        solve a suitable relaxation (R_t^{ij}) of the updated problem $P_t^i(x_{t-1}^{ik}, \psi_n^{i+1}, \xi_t^j)$ and collect cut coefficients (v_t^{ij}, π_t^{ij})
      end for
      add cut in the form of (3b) to ψ_{t-1}^i to get ψ_{t-1}^{i+1}
    end for
  end for
  /* Lower bound update */
  solve $P_1^i(\bar{x}_0, \psi_1^{i+1})$ and set LB to the optimal value
  $i \leftarrow i + 1$
end while

2) Lagrangian cut: This family of cuts is based on solving a Lagrangian dual of $P_t^i(x_{t-1}^i, \psi_t^{i+1}, \xi_t^j)$ by dualizing the redundant constraint (2c) in (2). In particular, the relaxation problem R_t^{ij} is defined as

$$(R_t^{ij}) \quad \begin{array}{ll} \max_{\pi_t} \min & g_t(x_t, y_t, z_t) + \psi_t^i + \pi_t^\top (x_{t-1}^i - z_t) \\ & \text{s.t.}(x_t, y_t, z_t) \in X_t(\xi_t^j) \\ & x_t \in \{0,1\}^d,\ z_t \in [0,1]^d. \end{array}$$

The key property of (R_t^{ij}) is that the Lagrangian dual has zero duality gap when the state variable x_t is binary [28]. This guarantees that the Lagrangian cuts generated from (R_t^{ij}) in the backward step are tight.

3) Strengthened Benders cut.: The Lagrangian dual (R_t^{ij}) may be difficult to solve to optimality. A heuristic is to solve the inner minimization in (R_t^{ij}) with a fixed dual solution obtained from the LP dual problem. This cut has the same coefficients as the Benders, but tighter constant term, therefore, can be stronger than Benders cut. We call it strengthened Benders cut.

4) Illustration of convergence behavior of cut families: Here we give a simple illustration of the typical convergence behavior of the above three families of cuts. Figure 1 shows the lower bound (red) and statistical upper bounds (green) and a 95% confidence interval. We can see that Benders (B) cuts never close the optimality gap, while the rest three Strengthened Benders (SB), Lagrangian (L), and SB+L all close the optimality gap. The figures also show the lower bounds usually converge quite fast, which is typical for SDDP-type algorithms. For more details, see [30, Chapter 3].

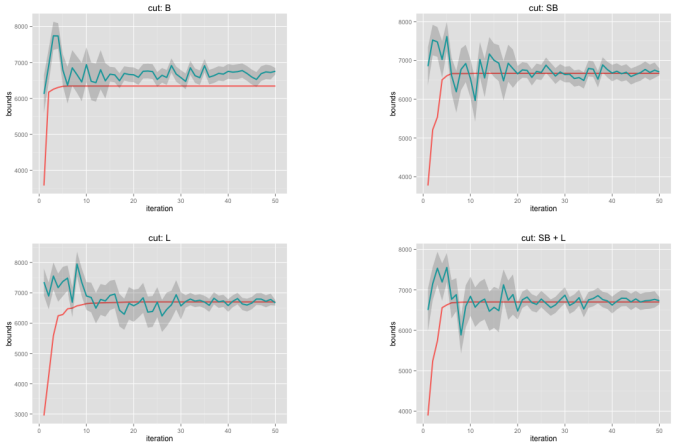

Fig. 1: Convergence behavior of different cut families.

III. Formulation of Multistage Stochastic UC

A. Problem Formulation

We first propose an MSUC formulation with uncertain net load modeled by a scenario tree. Due to special structures of this MSUC, the SDDiP algorithm cannot directly be applied. We then discuss techniques to reformulate it such that SDDiP can be used to solve it.

$$\min \sum_{n\in\mathcal{T}} p_n \left[\sum_{i\in\mathcal{G}} \left(\overline{S}_i u_{ni} + \underline{S}_i v_{ni} + f_i(y_{ni})\right) + C_p(\delta_n^+ + \delta_n^-)\right]$$

$$\text{s.t.} \textstyle\sum_{i\in\mathcal{G}} y_{ni} + \delta_n^+ - \delta_n^- = \sum_{b\in\mathcal{D}} D_{nb}, \forall n \in \mathcal{T} \quad (4a)$$

$$\textstyle\sum_{i\in\mathcal{G}} r_{ni} \geq R_{t(n)},\ \forall n \in \mathcal{T} \quad (4b)$$

$$\left|\textstyle\sum_{b\in\mathcal{B}} K_{\ell b}\left(\sum_{i\in\mathcal{G}_b} y_{ni} - D_{nb}\right)\right| \leq F_\ell, \forall \ell \in \mathcal{L}, n \in \mathcal{T} \quad (4c)$$

$$y_{ni} + r_{ni} \leq \overline{P}_i x_{ni},\ y_{ni} \geq \underline{P}_i x_{ni},\ \forall i \in \mathcal{G}, n \in \mathcal{T} \quad (4d)$$

$$y_{ni} - y_{a(n),i} \leq \Delta_i^{\text{SU}} u_{ni} + \overline{\Delta}_i x_{a(n),i},\ \forall i \in \mathcal{G}, n \in \mathcal{T} \quad (4e)$$

$$y_{a(n),i} - y_{ni} \leq \Delta_i^{\text{SD}} v_{ni} + \underline{\Delta}_i x_{ni},\ \forall i \in \mathcal{G}, n \in \mathcal{T} \quad (4f)$$

$$x_{ni} - x_{a(n),i} \leq u_{ni},\ \forall i \in \mathcal{G}, n \in \mathcal{T} \quad (4g)$$

$$x_{ni} - x_{a(n),i} = u_{ni} - v_{ni},\ \forall i \in \mathcal{G} \quad (4h)$$

$$\textstyle\sum_{m\in\mathcal{P}(n,UT_i-1)} u_{mi} \leq x_{ni},\ \forall i \in \mathcal{G}, n \in \mathcal{T} \quad (4i)$$

$$\textstyle\sum_{m\in\mathcal{P}(n,DT_i-1)} v_{mi} \leq 1 - x_{ni},\ \forall i \in \mathcal{G}, n \in \mathcal{T} \quad (4j)$$

$$x_{ni}, u_{ni}, v_{ni} \in \{0,1\}, y_{ni}, r_{ni} \geq 0,\ \forall i \in \mathcal{G}, n \in \mathcal{T}. \quad (4k)$$

In the above formulation, $x_{ni}, u_{ni}, v_{ni}, y_{ni}$ are respectively the commitment, start-up, shut-down binary decisions, and continuous dispatch decision of generator i in node n of the scenario tree, all of which are the state variables; r_{ni} is the reserve level of generator i at node n and δ_n^+, δ_n^- are slack variables in the energy balance constraint (4a), which are the local continuous variables. The objective function consists of the expectation of four terms, namely the start-up cost, the shut-down cost, the dispatch cost, and the penalty from electricity shortage or over generation. Energy balance constraint is

enforced in (4a) and the requirement for total reserved capacity at each stage in (4b). DC power flow equations are imposed by (4c). Constraints (4d) specify the output capacities for each generator. Ramping constraints are enforced by (4e)-(4f). Constraints (4g)-(4h) link the generator states with commitment decisions. Minimum up and down time constraints are enforced by (4i)-(4j).

B. Reformulation of State Variables

One obstacle of applying SDDiP directly to MSUC (4) is the minimum up and down time constraints (4i)-(4j), which couple state variables u_{ni}, v_{ni}'s from more than two stages, while SDDiP requires that the stage t problem depend only on the state variables in stage $t-1$. We propose a reformulation of state variables to resolve this problem. In particular, we create two sets of new variables at each node n: $\{u_{ni}^{(k)}, k = 0, \dots, UT_i - 1\}$ and $\{v_{ni}^{(k)}, k = 0, \dots, DT_i - 1\}$. Constraints (4i)-(4j) are then equivalent to the following set of inequalities:

$$\textstyle\sum_{k=0}^{UT_i-1} u_{ni}^{(k)} \le x_{ni}, \quad \sum_{k=0}^{DT_i-1} v_{ni}^{(k)} \le 1 - x_{ni}, \tag{5a}$$

$$u_{ni}^{(0)} - u_{ni} = 0, \quad v_{ni}^{(0)} - v_{ni} = 0, \tag{5b}$$

$$u_{ni}^{(k)} = u_{a(n),i}^{(k-1)}, \quad \forall k = 1, \dots, UT_i - 1 \tag{5c}$$

$$v_{ni}^{(k)} = v_{a(n),i}^{(k-1)}, \quad \forall k = 1, \dots, DT_i - 1. \tag{5d}$$

As a result, it is sufficient to pass $x_{a(n),i}$, $y_{a(n),i}$, $\{u_{a(n),i}^{(k)}\}_{k=0}^{UT_i-2}$, and $\{v_{a(n),i}^{(k)}\}_{k=0}^{DT_i-2}$ to node n, all of which comes from the parent node $a(n)$.

Moreover, SDDiP requires all state variables to be binary. In MSUC, generator states (x) and commitment decisions (u, v) are already binary, however, the dispatch decision (y) is continuous. To resolve this issue, we discretize the dispatch decision y using a binary approximation. In particular, a continuous state variable $y \in [0, U]$ can be approximated to any precision of $\epsilon \in (0, 1)$ as $y = \sum_{i=1}^{\kappa} 2^{i-1}\epsilon\lambda_i$ where $\lambda_i \in \{0, 1\}$ and $\kappa = \lfloor \log_2(U/\epsilon) \rfloor + 1$. Note that $|y - \sum_{i=1}^{\kappa} 2^{i-1}\epsilon\lambda_i| \le \epsilon$. The total number k of binary variables introduced to approximate the d state variables thus satisfies $k \le d(\lfloor \log_2(U/\epsilon) \rfloor + 1)$. Once the continuous state variables y_{ni} are binarized, the Lagrangian cuts introduced in Section II-D2 will be tight for approximating the ECGF function and the SDDiP algorithm is guaranteed to converge to global optimality of the MSUC problem (4).

IV. SDDiP Enhancements

In this section, we describe several enhancements to the basic SDDiP method described previously.

A. level method for Lagrangian Cut

To obtain the cut coefficients of the Lagrangian cuts, one needs to solve the Lagrangian dual problem (R_t^{ij}) for each problem in the forward step. This is a non-smooth convex optimization problem. A popular approach to solve such problems is the subgradient method (see e.g.,[31]). However, it is well known that the subgradient method can be very slow. We propose to use a stabilized cutting plane method, called the level method, to solve (R_t^{ij}). The level method dynamically constructs a convex piecewise linear underestimation function of the original objective using subgradient information, then solves this approximate problem and projects a minimizer of the model function to an appropriate level set so that its objective value lies in some neighborhood of the objective of the current iterate [32]. In this way the iterates are regularized and the method achieves a theoretical optimal convergence rate. It has also been proven to be very effective in practice [32]. In Section VI, we show the advantage of the level method over the basic subgradient algorithm for obtaining Lagrangian cut coefficients.

B. Hybrid Model using "Breakstage"

The quality of a policy obtained by SDDiP depends on the quality of the approximation of the ECTG in each stage. Intuitively, the stages further in the future has less influence on the decisions of earlier stages. Motivated by this, we propose a hybrid modeling approach, which allows us to improve the solution time while not compromising its quality to a large extent. This approximation relies on a prescribed stage t_b, which we will refer to as the *breakstage* hereafter. More specifically, in any decision stage before t_b, we solve the $P_t^i(x_{t-1}^i, \psi_t^{i+1}, \xi_t^j)$ in the forward step, where state variables are all binary after binarizing the dispatch decision y_{ni}'s. From stage t_b onward, we use the original continuous state variable y_{ni}'s without binarization. All three types of cuts at every stage are still valid, except that the Lagrangian cuts after t_b are not guaranteed to be tight.

In our experiments, we further relax all integrality constraints after t_b to improve solution time. As a result, only Benders cuts are added for stage problems after t_b. If $t_b = 0$, the method reduces to SDDP applied to the LP relaxation of the original formulation; if $t_b = T + 1$, the fully discretized problem is solved by SDDiP. The breakstage gives us the flexibility to evaluate the trade-off between solution time and solution quality. Solving the LP relaxation and using a state space of smaller dimension both contribute to the reduction of solution time. In addition, one can always adjust the policy if new information, e.g., a more accurate renewable generation forecast, becomes available.

C. Backward Parallelization

In the backward step of SDDiP, multiple scenario problems are solved, then the cut coefficients returned by each of them are aggregated to produce a cut for its previous stage problem. Since these scenario problems are independent from each other, we implement a parallelization scheme using the OpenMP API to speed up the backward step.

V. Experimental Settings

In this section, we discuss our experimental settings. The 14-bus system has 5 generators, 20 transmission lines, and 11 demand buses; and the 118-bus system includes 54 generators, 186 transmission lines, and 91 demand buses. Most data about the physical electrical network is from MatPower 6.0. Ramping limit is set to be 80% of the maximum generation capacity or

specified otherwise. Minimum up and down times vary from 1 to 10 hours. To avoid infeasibility, slack variables are added to the load balance constraints and penalized with a large cost in the objective function. All penalty costs are assumed to be \$5000 per MW.

A. Problem Scale and Intractability by Existing Methods

Deriving strengthened Benders cuts and Lagrangian cuts require solving MIPs in each stage. Therefore, the size of the stage problem greatly affects the solution time. In the 14-bus system, each nodal problem (2) has 127 binary state variables, 10 integer local variables, and 174 continuous local variables. For the 118-bus system, the corresponding numbers are 1086, 108, and 1514. The total number decision variables of the MSUC problem is the above numbers of variables in a nodal problem multiplied with the number of nodes in the scenario tree. In a scenario tree of 24 stages with 20 branches in each stage that we will test our algorithm on, the number of nodes is 20^{23}. Therefore, the total number of integer decisions in the 14-bus MSUC models is over 10^{32} and this number is over 10^{33} for the 118-bus MSUC model. Neither 14-bus nor 118-bus MSUC models can be solved by any of the existing methods due to their complexity introduced by integer decisions and their sheer size. To the best of our knowledge, the proposed sampling-based SDDiP algorithm is the first one that can solve such huge-scale MSUC problems to near-optimality as we will show below.

B. Scenario Tree Generation

To generate a recombining scenario tree, we start with a given net load in the first stage (12am, $t = 1$). At each following hour t, net load $\tilde{D}_t$ is generated from a nominal net load D_t according to $\tilde{D}_t = D_t \cdot \xi_t$, where $\xi_t \sim \Xi_t$, and Ξ_t is an estimated distribution from historical data. The total net load across the network is then allocated to each load bus according to the ratio implied from the base load profile, i.e., the proportion of the net load at each bus is the same for all realizations within the same stage.

For the 14-bus system, we assume ξ_t follows a uniform distribution $\mathcal{U}(1-\alpha, 1+\alpha)$ for all $t > 1$, and $\alpha \in [0.1, 0.3]$. Six types of scenarios trees are generated, each of them is characterized by net load variation (α) and the number of outcomes at each stage (β). The corresponding tree is denoted by $\mathcal{T}_{14}^{\alpha,\beta}$. In our experiments, we consider $\alpha = 0.1, 0.2, 0.3$ and $\beta = 10, 20$. For the 118-bus system, we use a truncated normal distribution, which is estimated based on data from California ISO website. We used the hourly net load forecast and the actual net load data across the entire California network in February 2017. The forecast is generated day ahead. For each hour, the distribution of forecast-to-actual ratio is approximated by a normal distribution. We assume $\xi_t \sim TN(\mu_t, k^2\sigma_t^2)$, where $TN(\mu_t, k^2\sigma_t^2)$ is the normal distribution $\mathcal{N}(\mu_t, k^2\sigma_t^2)$ truncated between $\mu_t \pm 3k\sigma_t$, and μ_t, σ_t are estimated from historical data. The scenario tree, denoted by $\mathcal{T}_{118}^{k,\beta}$, is then characterized by k and the number of outcomes at each stage (β). In our experiments, we fix $\beta = 20$ and consider k varying from 1.0 to 1.3. Note that $k = 1.3$ represents a situation with very significant uncertainty in net load. Figure 2 is an illustration of 50 independent scenarios from scenario tree $\mathcal{T}_{14}^{0.2,20}$(left) and $\mathcal{T}_{118}^{1.3,20}$ (right).

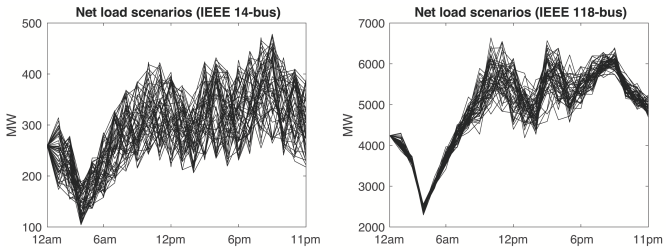

Fig. 2: 50 scenarios from scenario trees $\mathcal{T}_{14}^{0.2,20}$(left) and $\mathcal{T}_{118}^{1.3,20}$ (right).

C. Other Implementation Details

In the forward step of the SDDiP algorithm, we generate candidate solutions for five independent sample paths, and in the backward step, we evaluate two of them which result in the highest cost. The Lagrangian dual problem is solved to optimality using a basic subgradient algorithm and the level method with an optimality tolerance of 10^{-4} for the 14-bus system and 5×10^{-4} for the 118-bus system, respectively. Other relative MIP tolerance is set to be the same as above for each system. The SDDiP algorithm is implemented in C++ with CPLEX 12.7.0 to solve the MIP and LP subproblems. All experiments are performed on a 16-core machine with Intel Xeon E5-2630 v3 @2.40GHz CPUs and 128GB of main memory. Reported solution times are wall clock times.

VI. Computational Results

Our experiments consist of two phases: Phase I, run SDDiP and obtain a UC and dispatch policy on a scenario tree; Phase II, evaluate the obtained policy on a new tree. For this purpose, we generate two scenario trees for each type (i.e. for each fixed value of (α, β) of $\mathcal{T}_{14}^{\alpha,\beta}$ and (k, β) of $\mathcal{T}_{118}^{k,\beta}$), the first one used in Phase I and the other used for evaluation in Phase II.

In Phase I, we solve SDDiP with different breakstages (t_b). As mentioed earlier, when $t_b = 0$, SDDiP reduces to standard SDDP. If $t_b = 1$, nothing changes except that the first-stage problem becomes an MIP. When $t_b > 1$, other types of cuts may be added to the stage problems before $t_b - 1$. In particular, we consider five different cut combinations: Benders cut only (B), strengthened Benders cut only (SB), Lagrangian cut obtained by subgradient method (Sub), Lagrangian cut obtained by the level method (Level), and strengthened Benders cut plus Lagrangian cut obtained by the level method (SB + Level).

Once t_b and cut families are determined, SDDiP starts. In the first half of iterations, we ignore any integrality constraints and only turn on Benders cuts to get a rough estimation of the ECTG functions. In the second half, we restore these integrality constraints and add other types of cuts to improve the estimation. The final statistical upper bound is evaluated based on a set of 800 independent forward sample paths. SDDiP terminates after a fixed number of iterations.

In Phase II, we reinforce the integrality constraints in stage problems after t_b. A set of 800 scenarios is sampled independently from the second scenario tree, forward problems

are solved with the policy obtained in Phase I, and the cost associated with each scenario is recorded. The performance of the policy is evaluated by comparing the lower bound returned by SDDiP in Phase I, with the right endpoint of 95%-CI for the sample mean of scenario costs obtained from Phase II. All results in this section are averaged over 3 independent runs. We discuss our findings with respect to following three questions:

1) Which cut combination(s) perform the best in SDDiP?
2) What is the effect of different choices of breakstage?
3) What is the speed-up ratio and parallel efficiency from the backward parallelization?

A. Results for 14-bus MSUC

1) Cut Combinations: To test the power of different families of cuts, we solve each instance with breakstage $t_b = 25$, i.e., the fully binary problem. In the forward step, we solve MIPs to obtain binary candidate solutions, and in the backward step, different cuts are generated by evaluating these solutions. The power of each cut family is assessed based on SDDiP gap, solution time, and evaluation gap. The number of iterations in SDDiP is fixed at 150 for instances with $\alpha = 0.1, 0.2$ and 500 for instances with $\alpha = 0.3$. Figures 3-4 show the SDDiP results of the six instances with different cut combinations. Figure 3 presents the optimality gap between the final lower bound and statistical upper bound. Figure 4 contains the solution time of the SDDiP algorithm. The horizontal axis indicates the instances indexed by the (α, β) pair.

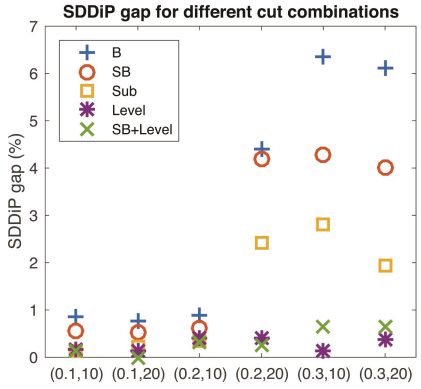

Fig. 3: SDDiP optimality gaps for different cut combinations.

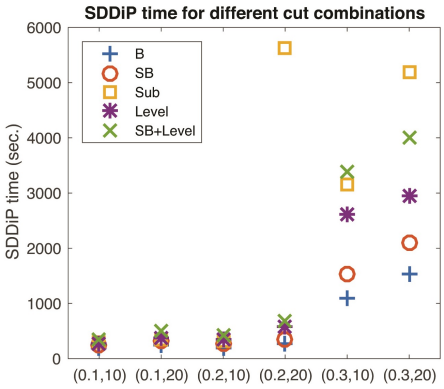

Fig. 4: SDDiP running times for different cut combinations.

Clearly, `SB+Level` and `Level` yield the smallest gap with a reasonable solution time among all. When the net load variation is small, using any type of these cuts is sufficient. When the variation becomes bigger, however, at least one family of tight cuts is needed to close the gap. Strengthened Benders cut slightly improves the SDDiP gap over only using Benders cut. Lagrangian cuts significantly improves the SDDiP gap over strengthened Benders cut. In addition, it is evident that the level method significantly outperforms the subgradient method in closing optimality gap with moderate increase in computation time.

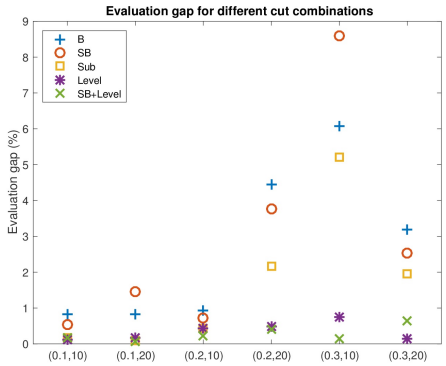

Fig. 5: Evaluation of policies obtained by different cut combinations.

The Phase II evaluation results are summarized in Figure 5. `SB+Level` and `Level` produce the most stable policies and yield the tightest statistic upper bound estimation. The policy approximated by Lagrangian cuts using subgradient method is again shown to be inferior to the one with the level method. In addition, we can observe a large evaluation gap for the policy characterized by the strengthened Benders cut in the instance $(0.3, 10)$. A possible reason is that 10 realizations per stage is not enough to represent the uncertainty with such big variation, the scenario tree used in the evaluation phase has some extreme scenarios not assessed in Phase I.

In summary, `SB+Level` or `Level` is the best cut combination for SDDiP, and solving the Lagrangian dual problem using the level method is more efficient and stable.

2) Effect of Breakstage: We next study the hybrid modeling approach proposed in Section IV-B. In particular, we choose 6 values for t_b, ranging from 0 to 25. When $t_b = 0$, the standard SDDP algorithm is used to solve the LP relaxation of the original problem. When $t_b > 1$, both strengthened Benders cuts and Lagrangian cuts (using the level method) are used in the backward step for stage problems before t_b. The number of iterations in SDDiP is fixed at 150 for instances with $\alpha = 0.1, 0.2$ and 500 for instances with $\alpha = 0.3$.

Figure 6 shows the optimality gap achieved in evaluation phase. We see that for instances with small net load variation ($\alpha \leq 0.2$), very small breakstages are sufficient to find very good solution with optimality gap less than 1%. When the uncertainty variation is high ($\alpha = 0.3$), optimality gap quickly decreases for $t_b \geq 12$. Figure 7 shows that the solution time for the SDDiP algorithm increases as the breakstage t_b increases. This is simply because more MIPs are solved as t_b increases.

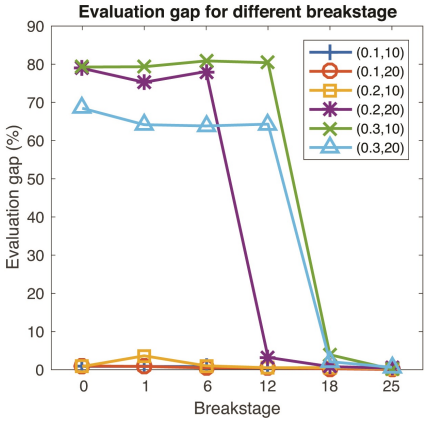

Fig. 6: Effect of breakstage on SDDiP optimality gap.

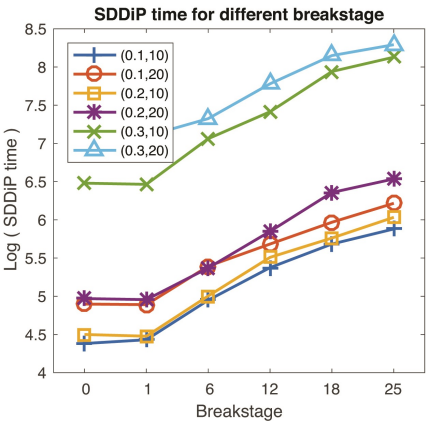

Fig. 7: Effect of breakstage on SDDiP running time.

3) Backward Parallelization: Let $T(k)$ be the solution time when k threads are used. We define *speed-up ratio* by $\frac{T(1)}{T(k)}$, and *efficiency* by $k\frac{T(1)}{T(k)}$. Figure 8 depicts an average speed-up ratio and efficiency graph with respect to the number of threads for a particular instance. We use 32 threads in all of our computation experiments. On average, the maximum speed-up ratio is 4.8 with an efficiency of 15%.

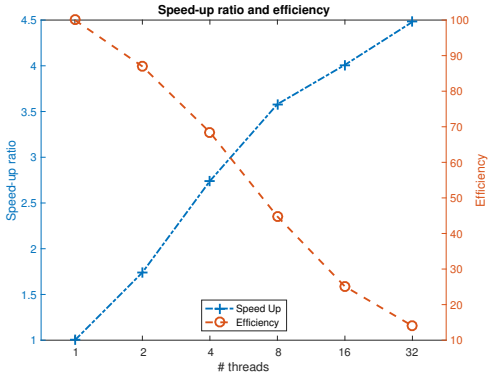

Fig. 8: Parallelization speed-up ratio & efficiency ($\mathcal{T}_{14}^{0.2,20}$ instance).

B. 118-bus Results

Similar to the 14-bus system, the experiments for the 118-bus system also consists of two phases: SDDiP and policy evaluation. We fix $\beta = 20$ in each scenario tree tested. An instance involves 20^{23} scenarios and over 10^{33} binary variables. Each instance is indexed by a pair (r, k), where r is ramping ratio with respect to the output capacity, and k is the parameter in the truncated normal distribution. We consider 12 instances with $r = 0.9, 0.8, 0.7$ and $k = 1.0, 1.1, 1.2, 1.3$. A smaller r indicates more restricted ramping constraints, while a larger k value suggests a more volatile scenario tree. Here, we do not consider k bigger than 1.3 because a larger value incurs a net load which exceeds the system's total generation capacity.

TABLE I: Computational results for 118-bus system

Instance (r, k)	Time (sec.)	Eval. Gap (%)	Instance (r, k)	Time (sec.)	Eval. Gap (%)
(0.9, 1.0)	4389	[0.47, 0.68]	(0.8, 1.2)	4424	[0.51, 0.75]
(0.9, 1.1)	4387	[0.51, 0.59]	(0.8, 1.3)	4455	[0.55, 0.96]
(0.9, 1.2)	4394	[0.50, 0.77]	(0.7, 1.0)	4389	[0.37, 0.63]
(0.9, 1.3)	4405	[0.55, 0.69]	(0.7, 1.1)	4427	[0.58, 0.84]
(0.8, 1.0)	4333	[0.48, 0.63]	(0.7, 1.2)	4455	[0.50, 1.12]
(0.8, 1.1)	4362	[0.48, 0.58]	(0.7, 1.3)	4521	[0.67, 1.28]

Table I contains the SDDiP computation time and evaluation results for the 118-bus system. The results indicate using SDDiP with Benders cut only is sufficient to produce an accurate (0.47%-1.28% gap) within 1.25 hours. This could be due to the tight formulation of a single scenario deterministic UC problem. To verify the tightness of the LP relaxation gap, we independently generate 100 scenarios from the most volatile load distribution ($k = 1.3$), and solve a deterministic 24-hour UC problem and its LP relaxation for each of the scenarios. The ramping limit is set to be 70% of the maximum generation capacity. Indeed, the average LP gap over these 100 instances is only 0.254%. Given that our uncertainty variation is based on real data, such a small LP relaxation gap suggests that the SDDiP with standard Benders cut is good enough to solve this large-scale MSUC instance.

VII. Conclusion

In this paper, we propose a stagewise-decomposition algorithm based on SDDiP with various algorithmic enhancements to solve large-scale MSUC problems. Systematic numerical experiments demonstrate that the proposed algorithm can successfully handle MSUC problems with a huge number of scenarios that were impossible for existing methods. It is also verified that Lagrangian cuts are indispensable in achieving exact solution and convergence. Our experiments show that when solving the Lagrangian relaxation of the stage problem, the level method performs superior to the standard subgradient method. We also observe that for the 118-bus system, it suffices to use SDDiP with only standard Benders cuts to obtain a good policy.

There are several interesting future research questions related to MSUC. In this paper, we decompose the 24-hour MSUC

problem on an hourly basis. An alternative is to consolidate several consecutive hours into one stage. Such a formulation increases the size of a stage problem but reduces the total number of decision stages. It would be interesting to investigate how such an aggregated model compares to the hourly based multistage model. Another direction is to study the MSUC problem under a risk-averse setting, as system reliability is of utmost importance in practice.

REFERENCES

[1] M. Tahanan, W. Van Ackooij, A. Frangioni, and F. Lacalandra, "Large-scale unit commitment under uncertainty," *4OR*, vol. 13, no. 2, pp. 115–171, 2015.

[2] W. Van Ackooij, I. D. Lopez, A. Frangioni, F. Lacalandra, and M. Tahanan, "Large-scale unit commitment under uncertainty: An updated literature survey," Universita di Pisa, Tech. Rep., 2018.

[3] Q. P. Zheng, J. Wang, and A. L. Liu, "Stochastic optimization for unit commitment—a review," *IEEE Trans. on Power Syst.*, vol. 30, no. 4, pp. 1913–1924, 2015.

[4] A. I. Mahmutogullari, S. Ahmed, O. Cavus, and M. S. Akturk, "The value of multi-stage stochastic programming in risk-averse unit commitment problems," 2018, working paper (presented at the 2018 INFORMS Optimization Society Conference).

[5] P. Carpentier, G. Gohen, J.-C. Culioli, and A. Renaud, "Stochastic optimization of unit commitment: a new decomposition framework," *IEEE Trans. on Power Syst.*, vol. 11, no. 2, pp. 1067–1073, 1996.

[6] S. Takriti, B. Krasenbrink, and L. S.-Y. Wu, "Incorporating fuel constraints and electricity spot prices into the stochastic unit commitment problem," *Oper. Research*, vol. 48, no. 2, pp. 268–280, 2000.

[7] K. C. Kiwiel, "An aggregate subgradient method for nonsmooth convex minimization," *Math. Prog.*, vol. 27, no. 3, pp. 320–341, 1983.

[8] M. P. Nowak and W. Römisch, "Stochastic lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty," *Annals of Oper. Research*, vol. 100, no. 1-4, pp. 251–272, 2000.

[9] L. Bacaud, C. Lemaréchal, A. Renaud, and C. Sagastizábal, "Bundle methods in stochastic optimal power management: A disaggregated approach using preconditioners," *Computational Optimization and Applications*, vol. 20, no. 3, pp. 227–244, 2001.

[10] T. Shiina and J. R. Birge, "Stochastic unit commitment problem," *International Transactions in Operational Research*, vol. 11, no. 1, pp. 19–32, 2004.

[11] S. Takriti, J. R. Birge, and E. Long, "A stochastic model for the unit commitment problem," *IEEE Trans. on Power Syst.*, vol. 11, no. 3, pp. 1497–1508, 1996.

[12] L. Wu, M. Shahidehpour, and T. Li, "Stochastic security-constrained unit commitment," *IEEE Trans. on Power Syst.*, vol. 22, no. 2, pp. 800–811, 2007.

[13] T. Schulze, A. Grothey, and K. McKinnon, "A stabilised scenario decomposition algorithm applied to stochastic unit commitment problems," *Rapport technique, The University of Edinburgh, School of Mathematics*, 2015.

[14] D. Rajan and S. Takriti, "Minimum up/down polytopes of the unit commitment problem with start-up costs," *IBM Res. Rep*, 2005.

[15] J. Ostrowski, M. F. Anjos, and A. Vannelli, "Tight mixed integer linear programming formulations for the unit commitment problem," *IEEE Trans. on Power Syst.*, vol. 27, no. 1, pp. 39–46, 2012.

[16] G. Morales-España, J. M. Latorre, and A. Ramos, "Tight and compact milp formulation for the thermal unit commitment problem," *IEEE Trans. on Power Syst.*, vol. 28, no. 4, pp. 4897–4908, 2013.

[17] K. Pan and Y. Guan, "Strong formulations for multistage stochastic self-scheduling unit commitment," *Oper. Research*, vol. 64, no. 6, pp. 1482–1498, 2016.

[18] R. Jiang, Y. Guan, and J.-P. Watson, "Cutting planes for the multistage stochastic unit commitment problem," *Math. Prog.*, vol. 157, no. 1, pp. 121–151, 2016.

[19] M. V. Pereira and L. M. Pinto, "Multi-stage stochastic optimization applied to energy planning," *Math. Prog.*, vol. 52, no. 1-3, pp. 359–375, 1991.

[20] T. Rotting and A. Gjelsvik, "Stochastic dual dynamic programming for seasonal scheduling in the norwegian power system," *IEEE Trans. on Power Syst.*, vol. 7, no. 1, pp. 273–279, 1992.

[21] T. Archibald, C. Buchanan, K. McKinnon, and L. Thomas, "Nested benders decomposition and dynamic programming for reservoir optimisation," *Journal of the Operational Research Society*, vol. 50, no. 5, pp. 468–479, 1999.

[22] A. Shapiro, W. Tekaya, J. P. da Costa, and M. P. Soares, "Risk neutral and risk averse stochastic dual dynamic programming method," *European journal of operational research*, vol. 224, no. 2, pp. 375–391, 2013.

[23] S. Cerisola, Á. Baíllo, J. M. Fernández-López, A. Ramos, and R. Gollmer, "Stochastic power generation unit commitment in electricity markets: A novel formulation and a comparison of solution methods," *Oper. Research*, vol. 57, no. 1, pp. 32–46, 2009.

[24] J. Wang, J. Wang, C. Liu, and J. P. Ruiz, "Stochastic unit commitment with sub-hourly dispatch constraints," *Applied energy*, vol. 105, pp. 418–422, 2013.

[25] Q. P. Zheng, J. Wang, P. M. Pardalos, and Y. Guan, "A decomposition approach to the two-stage stochastic unit commitment problem," *Annals of Oper. Research*, vol. 210, no. 1, pp. 387–410, 2013.

[26] S. Cerisola, J. M. Latorre, and A. Ramos, "Stochastic dual dynamic programming applied to nonconvex hydrothermal models," *European Journal of Operational Research*, vol. 218, no. 3, pp. 687–697, 2012.

[27] G. Steeger and S. Rebennack, "Dynamic convexification within nested Benders decomposition using Lagrangian relaxation," *European Journal of Operations Research*, pp. 669–686, 2017.

[28] J. Zou, S. Ahmed, and X. A. Sun, "Stochastic dual dynamic integer programming," *Mathematical Programming*, 2018 (to appear).

[29] A. Shapiro, "Analysis of stochastic dual dynamic programming method," *European Journal of Operational Research*, vol. 209, no. 1, pp. 63–72, 2011.

[30] J. Zou, "Large scale multistage stochastic integer programming with applications in electric power systems."
[31] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," *lecture notes of EE392o, Stanford University, Autumn Quarter*, vol. 2004, 2003.
[32] C. Lemaréchal, A. Nemirovskii, and Y. Nesterov, "New variants of bundle methods," *Math. Prog.*, vol. 69, no. 1, pp. 111–147, 1995.