

State-space Approximate Dynamic Programming for Stochastic Unit Commitment

Zhang, W.; Nikovski, N.

TR2011-046 August 2011

Abstract

It is known that unit commitment problems with uncertainties in power demands and the outputs of some generators can be represented as factored Markov decision process models. In this paper we propose a state space approximate dynamic programming algorithm to solve such models. The algorithm features a method to generate representative system configurations (states) and a functional metric to measure the similarity among system configurations. Experimental results show that the algorithm outperforms two deterministic approaches in resulting in both lower risks and operational costs, and that it can solve larger problems than a stochastic approach based on decision space approximate dynamic programming.

North American Power Symposium (NAPS)

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

State-space Approximate Dynamic Programming for Stochastic Unit Commitment

Weihong Zhang and Daniel Nikovski, {wzhang, nikovski}@merl.com, phone 617-621-7516

Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, MA 02139, USA, fax 617-621-7550

Abstract—It is known that unit commitment problems with uncertainties in power demands and the outputs of some generators can be represented as factored Markov decision process models. In this paper we propose a state space approximate dynamic programming algorithm to solve such models. The algorithm features a method to generate representative system configurations (states) and a functional metric to measure the similarity among system configurations. Experimental results show that the algorithm outperforms two deterministic approaches in resulting in both lower risks and operational costs, and that it can solve larger problems than a stochastic approach based on decision space approximate dynamic programming.

I. INTRODUCTION

A *unit commitment (UC)* problem is concerned with finding optimal schedules for generators and their generations in order to meet user demands over a time horizon in observance of a number of operational constraints. A *stochastic unit commitment* problem extends the conventional UC formulation in that it admits and explicitly models the various sources of uncertainties such as demand fluctuations, mechanic errors in generator operations, varying wind velocities, intermittency of solar energy devices, etc. This paper intends to address two types of uncertainties, namely user demands and the generation outputs of the renewables. It is known that user demand cannot be precisely predicted because it varies with seasons in a year, calendar days, overall climate, user consumption patterns, and many other factors. In addition, the generations of the renewables are highly volatile. For instance, the generation of a wind turbine varies with several variables such as the rated maximum power, cut-in and cut-out speed, generator efficiency, air density, and wind velocities.

One consequence of introducing uncertainties to a UC formulation is that the conventional UC solution methods, based on dynamic programming, Lagrange relaxation, mixed integer programming, etc. [1], cannot be directly used to solve stochastic problems. The reason is that these methods solve a UC problem for *one realized instance* of the uncertain variables like user demand and the generation output, whereas a stochastic UC problem requires a solution method

to optimize the operating costs for *different realizations* of the uncertain variables and weight them by the probabilities of the occurrence of the realizations. Therefore, introduction of uncertainties to a UC problem adds extra computational complexity to its solution methods that have to solve an already difficult combinatorial optimization problem, despite that it does increase the expressiveness of the formulation model. Hence, one important direction in stochastic UC is to develop efficient solution algorithms to find optimal or sub-optimal schedules.

One such a proposal in [2] exploited a tree structure to organize the possible realizations (called *scenarios*) of a UC system. The scenarios tree is evaluated by a technique related to Lagrangian relaxation. However, the tree structure is limited to only a few scenarios. Recently [3] proposed to use factored Markov decision processes (fMDPs) to model a stochastic UC problem where the stochasticity in user demands and the outputs of non-dispatchable generators such as wind turbines is represented as relative probability distributions. These distributions potentially open a way to model an infinite number of possible system realizations. To address the solution complexity, [3] explored a decision-space approximate dynamic programming method (DSADP) that computes a set of representative commitment schedules (on/off decisions) of generators by varying the target demands, and organizes and evaluates them through an AND/OR tree data structure. Experiments showed that the fMDP method achieves a better trade-off between the costs and operating risk than the conventional approaches. However, the AND/OR-tree grows exponentially in the decision horizon and thus is impractical for long-horizon applications.

In this paper, we propose a state-space approximate dynamic programming (SSADP) algorithm for fMDP models. Here, a state (or configuration) consists of generators' commitment schedules, their operational times, the realization of the outputs of the non-dispatchable generators, and the realizations of the demand. In order to handle the huge state space, the SSADP method works with a reduced set of representative system

states, and uses a state aggregation approach with a suitable metric to represent the value function of states outside of this set. The size of the set is polynomial in the number of generators and the decision horizon, thus eliminating the limitation of the DSADP method. The representative states are generated by solving multiple instances of the economic dispatch problem for various levels of net power demand, which typically reduces to ordinary quadratic programming (QP). Our experiments show that, similarly to the DSADP method, the SSADP method outperforms several popular deterministic methods in obtaining solutions with lower cost and risk; and, furthermore, the SSADP method can solve much larger problems than the DSADP method.

II. BACKGROUND

Let N (K) be the number of available dispatchable (undispatchable) generator units, and T be the planning horizon, in suitable units (typically, one hour). The objective function of a stochastic unit commitment problem, presented in Equation II.1, is to minimize the total cost comprising of the generation costs $\sum f_i(x_t^i, u_t^i, y_t, d_t)$ of meeting demand d_t , the generator switching costs $\sum h_i(x_t^i, u_t^i, u_{t+1}^i)$ of changing generator states to u_{t+1}^i , and the penalty costs $g_t(u_t, y_t, d_t)$ resulted from failing to meet the demand d_t . Here, $u_t^i \in \{0, 1\}$ represents the commitment status (on or off) for unit i at step t , x_t^i represents the number of time steps that unit i has been on/off, d_t is the realizations of the random demand D_t ($1 \leq t \leq T$) from a known stochastic process, and y_t^k is the output of the k -th non-dispatchable generator from known stochastic processes. The configuration and the operational times of all controllable units at time t are respectively $u_t = [u_t^1, u_t^2, \dots, u_t^N]$ and $x_t = [x_t^1, x_t^2, \dots, x_t^N]$. Similarly, $y_t = [y_t^1, y_t^2, \dots, y_t^K]$. In addition, $\mathbf{E}_{u_0, x_0, y_0, d_0}$ denotes the expectation operator with regard to the initial configuration u_0 , operational time x_0 , the initial demand d_0 , and the initial output y_0 .

$$J^* = \min_{u_1, u_2, \dots, u_T} \mathbf{E}_{u_0, x_0, y_0, d_0} \left\{ \sum_{t=0}^{T-1} \left[\sum_{i=1}^N f_i(x_t^i, u_t^i, y_t, d_t) + \sum_{i=1}^N h_i(x_t^i, u_t^i, u_{t+1}^i) + g_t(u_t, y_t, d_t) \right] \right\} \quad (\text{II.1})$$

Note that by adding the operating cost and risk compensation cost together, the objective above represents a trade-off between fuel costs and risk. The penalty function $g_t(u_t, y_t, d_t)$ denotes the equivalent cost of the risk of being unable to meet demand d_t under the output y_t and the configuration u_t . It is proportional to the probability that the total capacity of the committed units in u_t plus the generation y_t of the non-dispatchable generators is less than the demand d_t . The function is defined in Equation II.2 where cap^i denotes the

maximum capacity of generator i , and the coefficient α specifies the relative preference between minimizing operating cost and risk of failure to meet demand.

$$g_t(u_t, y_t, d_t) = \alpha \Pr \left(\sum_{i=1}^N u_t^i cap^i + \sum_{k=1}^K y_t^k < d_t \right) \quad (\text{II.2})$$

A UC problem has to observe several constraints in minimizing the total cost. The load balance constraint states that the total generation of the the dispatchable and non-dispatchable generators must be equal to the demand at any time step. The positive excess spinning reserve constraint indicates that the total committed spinning capacity should be greater than the sum of the load and the required spinning reserve specified by regulators. Other constraints are concerned with the minimum and maximum generation capacity of the units, minimum up and down time constraints, and unit availability.

As shown in [3], the stochastic unit commitment problem with demand fluctuation and the generation variations of the non-dispatchable units can be described by means of a *factored Markov decision process* (fMDP). The model is *factored* because the system states or decisions are the cross-products of the states or decisions of the individual generators. The fMDP model components are specified as follows.

State: The system state at time t is represented as (u_t, x_t, y_t, d_t) where d_t is the instantiated demand, and y_t is the outputs of non-dispatchable generators.

Action: An action u_t at the system level is its intended commitment status of all generators at the next time step.

State transition: Given the current state (u_t, x_t, y_t, d_t) and an action u_{t+1} , the transition probability $\Pr(u_{t+1}, x_{t+1}, y_{t+1}, d_{t+1} | u_t, x_t, y_t, d_t)$ is factorized to $\Pr(u_{t+1}, x_{t+1} | u_t, x_t) \Pr(d_{t+1} | d_t) \prod_{k=1}^K \Pr(y_{t+1}^k | y_t^k)$. If u_{t+1} achieves its intended commitment status of all generators, $\Pr(u_{t+1}, x_{t+1} | u_t, x_t)$ is 1.0; it is 0.0 otherwise (impossible transition) [4]. The transition probabilities $\Pr(D_{t+1} = d_{t+1} | D_t = d_t)$ and $\Pr(Y_{t+1}^k = y_{t+1}^k | Y_t^k = y_t^k)$ are respectively for demands and the generation of non-dispatchable generators.

Cost: Given an MDP state (u_t, x_t, y_t, d_t) and an action u_{t+1} , the immediate one-step cost $c(u_t, x_t, u_{t+1}, y_t, d_t)$ is

$$c(u_t, x_t, u_{t+1}, y_t, d_t) = \sum_{i=1}^N f_i(x_t^i, u_t^i, y_t, d_t) + \sum_{i=1}^N h_i(x_t^i, u_t^i, u_{t+1}^i) + g_t(u_t, y_t, d_t) \quad (\text{II.3})$$

The fuel costs $f_i(x_t^i, u_t^i, y_t, d_t)$ are computed by solving the economic dispatch (ED) problem of minimizing $\sum_i F_i(p_i^j)$ where $F_i(p_i^j)$ is the cost of producing p_i^j units of electricity by generator i , subject to the generation limits for all generators

and the load balance constraint for the realization of the non-dispatchable variables y_t and demand d_t . Typically, the cost function $F_i(p_i^i)$ is quadratic in p_i^i , and the ED can be solved by means of quadratic programming (QP). The ED calculates the optimal generation amounts p_i^i of the committed units so that the cost of generation is minimized.

The straightforward solution to an fMDP is to expand the factored states and solve the resulting flat MDP by means of dynamic programming. It starts from the terminal step with the cost-to-go function $J_T(u_T, x_T, y_T, d_T) = 0$, and proceeds backwards to the first step by using the Bellman Equation (Equation II.4)[5]. The cost-to-go function $J_0(u_0, x_0, y_0, d_0)$ of the initial state corresponds to the minimum in Equation II.1: $J^* = J_0(u_0, x_0, y_0, d_0)$. Obviously, this full state space sweeping strategy is impractical. Therefore existing methods have been working on an approximate (reduced) decision and/or state space but attempted to achieve high-quality solutions [6]. One such a method is DSADP [3]. It precalculated a collection of schedules (i.e., decisions), which approximates the decision space, for a set of demands and alternated the schedules for different demand realizations. The decisions and the demand realizations are organized in an AND/OR tree structure [7]. However, the tree grows exponentially in the planning horizon in the worse case of complexity.

$$J_t(u_t, x_t, y_t, d_t) = \min_{u_{t+1}} \{c(u_t, x_t, u_{t+1}, y_t, d_t) + \sum_{d_{t+1}, y_{t+1}} Pr(d_{t+1}, y_{t+1} | d_t, y_t) J_{t+1}(u_{t+1}, x_{t+1}, y_{t+1}, d_{t+1})\} \quad (\text{II.4})$$

III. STATE SPACE APPROXIMATE DYNAMIC PROGRAMMING

This section presents our state space approximate dynamic programming algorithm (SSADP) in details.

A. Algorithm overview

A system state or a configuration (u_t, x_t, y_t, d_t) in the fMDP is comprised of four elements – the commitment status, the operating times, the output of non-dispatchable generators, and the user demand. Hence, the system state space grows exponentially with the generators' number. To provide computationally efficient solutions, our SSADP method takes two steps to reduce the state space. At the first step, the SSADP algorithm aggregates the output of the non-dispatchable generators into the demand variable, by subtracting these outputs from the total demand to arrive at the net demand. The aggregation reduces the state space by one factoring variable. Nonetheless, this reduction does not change the exponential growth of the reduced state space in the number of the generating units. At the second step of reducing the state space,

the SSADP method selects a collection of target demands by varying the target reserves. For every demand, a deterministic UC problem can be solved via any deterministic approaches to obtain its commitment schedules and system states. These schedules and configurations can provide representative options to the stochastic UC problem as follows – for any demand realization, the SSADP method can choose a schedule from the representative set that is optimal or suboptimal in the deterministic sense. In addition, if the demand realization follows a different pattern from the sampled demands, in time the demand realization may pick a schedule resulting from one target demand at one time step, and switch to another schedule resulting from another target demand at the next time step. This schedule switching feature signifies that a stochastic planner is conditional on the realized demands.

After the system schedules and states are generated, the SSADP method seeks to perform value updates over them. One issue arises when the reachable configurations of a given state do not belong to the generated state space. The SSADP features a similarity metrics quantifying the similarity among states/configurations. When a state's value is not available, the value of its most similar state is used instead.

Figure III.1 illustrates how SSADP works. It shows two time steps at t and $t+1$. The two large solid eclipses represent the full state space. The target demands $D_1 \sim D_4$ are selected to generate the system configurations, denoted by plus signs. The set of the plus signs is the generated state space that SSADP works with. When performing value updates for state s_2 , the two reachable states s_5 and s_6 , represented as stars, are not calculated at Step $t+1$. The SSADP uses the similarity metric to search the most similar states. Found are those represented by pluses at Step $t+1$ and enclosed in the small dashed eclipses of states s_5 and s_6 . The values of the most similar states are used for the un-calculated states s_5 and s_6 .

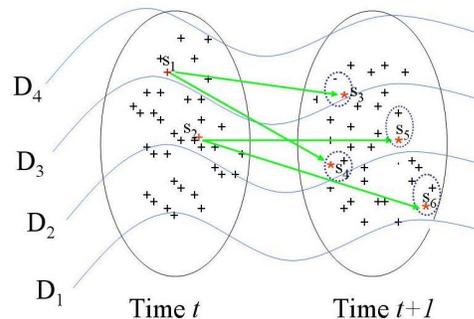


Figure III.1. State-Space Approximate dynamic programming (SSADP)

B. State space generation

The complete procedure of generating the state space consists of three steps – generating a set of target demands, solving optimal configuration problems to obtain their solution schedules, and adding the maximum system capacity to the set of the states and schedules.

Step 1. Generate demands $D_\beta = (1 + \beta)\bar{D}$ for β in $[\underline{\beta}, \bar{\beta}]$ where β is a target reserve level, \bar{D} is the expected demand, and $[\underline{\beta}, \bar{\beta}]$ ($\underline{\beta}(\bar{\beta})$ is a negative (positive) number) represents the range within which the target reserve level can vary. If β is uniformly sampled from the range, the demands constitute a uniform coverage of the demand space. The notation D_β is viewed as a vector of demands over time periods. Let D_t^β be its component at Step t .

Step 2. For each demand D_t^β , an *optimal configuration* problem is formulated to minimize the production cost of meeting the demand. Its objective function is the production cost at Step t . For thermal generators, the production cost is quadratic in the production amount p , so the formulated problem can be solved using quadratic program (QP). The QP only considers one single time step t ; therefore it cannot accommodate the temporal constraints and the transitional costs. The constraints are to abide by the generators' minimum/maximum capacity and the load constraint. Specifically, for each demand D_t^β , N (the number of the generators) QPs are formulated. In the i -th QP, it is mandated that the i -th generator is off, while others are on. This is the input configuration to the QP.

If the QP is infeasible (for example, the committed capacity is smaller than the demand), it does not introduce useful configurations. Otherwise, the QP solution schedule is examined for the on-generators in its input schedule. If the production of a generator is strictly positive, its commitment status remains unchanged; otherwise its status is set back to off in order to reduce excessively committed capacities even if its commitment status is on in the input configuration. The adjusted schedule is the output of the QP. The output schedule and the demand together form a configuration and is added to the set S_t^β . It is noted that any open source or commercial software package can be used to solve the QPs.

Let the notation S_t^β denote the set of schedules based on the demand D_t^β . After the schedules are identified by varying the demands and the time steps, the sets $\{S_t^\beta | \beta \in [\underline{\beta}, \bar{\beta}]\}$ contain the configurations for the demand D_β at time t ($\in [1, \dots, T]$).

It is worthy to note that the optimal configuration problem does not consider the temporal constraints such as the up/down times. This is mainly due to efficiency consideration – the

number of the variables is equal to the number of committed generators at one step. Certainly, the optimal configuration problem can be set up to observe the temporal constraints. In so doing, the number of the variables in a QP is equal to the total number of the committed generators over all time steps. If the decision has to be made along a long horizon, the computational cost of solving the optimal configuration problem itself would be rather high. However, one consequence of ignoring the temporal constraints is that the configurations calculated do not have the component of the operational times. To make them valid states, in the calculated configurations, the committed (uncommitted) generators are argumented with an operating time +1 (-1). This is an approximation of operational times to the real-world scenarios. The approximation is justified by the fact that the majority of the startup and shutdown costs of a power plant is caused mainly by switching on/off the generators rather than running them in the on/off status persistently.

Step 3. To complete the set S_t^β , a special configuration in which all generators are set to on and its demand is the total system generation capacity is added to S_t^β where $\bar{\beta}$ is the maximum β . Such a configuration represents the maximum demand that the system can meet when all generators are turned on and generate according to their maximum capacity. Obviously, when a demand realization exceeds the system generation capacity, a blackout occurs.

The number of the states generated above is polynomial in the number N of the generators. In fact, since **Step 2** generates at most $N_\beta * N$ configurations for one decision step where N_β is the number of selected β s and **Step 3** adds one more configuration, the total number of states for one decision step is at most $(N_\beta * N + 1)$. The overhead of obtaining these states is solving $N_\beta * N$ QPs.

C. Approximate dynamic programming

In DP updates over the reduced state space, the decision choices can be implicitly derived from the configurations in the approximate state space. Recall that since each system configuration specifies the generators' on/off status, it essentially prescribes action choices for all generators. Collectively, the action component of the configurations determines a decision space. This is the decision space that will be used in the SSADP updates. Therefore, the number of decision choices is polynomial in the number of generators since the number of decisions is not larger than the number of configurations at any time step (different system configurations may result in the same machine configurations due to different demands).

Since the full decision space is exponential in the number of generating units, the SSADP considers a much smaller decision set.

To make the decision choices explicit, a DP update is formulated in Equation III.1 where $u_{t+1}^{\beta'}$ is the generators' on/off status of a configuration in the union $\bigcup_{\beta} S_{t+1}^{\beta}$, and s_t^{β} is a system state/configuration generated for demand $(1 + \beta)\bar{D}$ at time t . Note that given a state s_t^{β} , its next actions $u_{t+1}^{\beta'}$ can be any one from the union $\bigcup_{\beta} S_{t+1}^{\beta}$. That is why the equation uses two different notations β and β' .

$$V_t(s_t^{\beta}) = \min_{u_{t+1}^{\beta'}} \{R(s_t^{\beta}, u_{t+1}^{\beta'}) + \sum_{s'} Pr(s'|s_t^{\beta}, u_{t+1}^{\beta'}) V_{t+1}(s') + \alpha G(s_t^{\beta}, u_{t+1}^{\beta'})\} \quad (\text{III.1})$$

The main issue in SSADP is lack of the values of the configurations that are not generated in the reduced state space. Specifically, in Equation III.1, given a state s_t^{β} and an action $u_{t+1}^{\beta'}$, the value $V_{t+1}(s')$ of the next state s' is unavailable if s' does not belong to the set of the generated configurations. The SSADP method proposes a similarity metric to address this issue. Given two states/configurations s and s' , the metric $\tau(s, s')$ measures the similarity between them. When the value $V_{t+1}(s')$ is undetermined, the value of the state that minimizes the metric $\tau(s, s')$ is used. Such a state is *the most similar state* to s' . That is, if $s^* = \min_s \tau(s, s')$, then $V_{t+1}(s') = V_{t+1}(s^*)$.

Clearly, the SSADP performance relies on the metric that measures the similarity of two states. From a cost perspective, if the cost-to-go value $V_{t+1}(s_1)$ is closer to $V_{t+1}(s_2)$, state s_1 is more similar to s_2 . However, these costs are not known unless the DP step is solved exactly. The principle in designing a metric is that the metric should mimic the cost to some extent, and meanwhile calculating the metric should be computationally inexpensive. Following these principles, we propose a metric that takes into account the committed capacity, the transitional capacity, the demand difference, and a risk-related component. Let us write states $s_1 = (u_1^1, u_1^2, \dots, u_1^N, x_1^1, x_1^2, \dots, x_1^N, d_1)$ and $s_2 = (u_2^1, u_2^2, \dots, u_2^N, x_2^1, x_2^2, \dots, x_2^N, d_2)$ and define the metric $\tau(s_1, s_2)$ as follows.

- The Committed Capacity (*CC*) of a state (say s_1) is the sum of the maximum capacities for those *on*-generators, i.e. $CC(s_1) = \sum_{i=1}^N u_1^i cap^i$. The Committed Capacity Difference (*CCD*) of two states s_1 and s_2 is the difference of $CC(s_1)$ and $CC(s_2)$, i.e., $CCD(s_1, s_2) = |CC(s_1) - CC(s_2)|$. The *CCD* measures the similarity of two states in meeting the demand up to their maximum committed capacity. Smaller *CCD* means more similar states.
- The Transitional Capacity (*TC*) of two states s_1 and s_2

is the sum of the capacities for those generators that are on in one state and off in the other, i.e., $TC(s_1, s_2) = \sum_{u_1^i + u_2^i = 1} cap^i$. The *TC* is 0.0 if two states have the same commitment schedules. The *TC* emulates the transition cost from one state to the other.

- The Demand Difference (*DD*) of two states s_1 and s_2 is the difference of their associated demands, i.e., $DD(s_1, s_2) = |d_1 - d_2|$. The *DD* emulates the cost difference when the system meets different demands.
- The Risk-related Cost Difference (*RCD*) of two states s_1 and s_2 is the difference of their cost associated with risk, i.e., $RCD(s_1, s_2) = |risk(s_1) - risk(s_2)|G$ where $risk(s)$ is the stepwise risk determined by state s (calculated as before), and G is an adjustable positive number. Large G value represents more penalty in the cost.

The overall similarity of two states s_1 and s_2 is defined as a weighted sum of the *CCD*, *TC*, *DD*, and *RCD*, i.e.,

$$\tau(s_1, s_2) = \gamma_1 * CCD(s_1, s_2) + \gamma_2 * TC(s_1, s_2) + \gamma_3 * DD(s_1, s_2) + \gamma * RCD(s_1, s_2) \quad (\text{III.2})$$

where γ_1 , γ_2 , γ_3 and γ are parameters to adjust the weights of different components in the metric. Such a metric is non-negative in that $\tau(s_1, s_2) \geq 0.0$, identical of indiscernibles in that $\tau(s_1, s_2) = 0$ if and only if $s_1 = s_2$, and symmetrical in that $\tau(s_1, s_2) = \tau(s_2, s_1)$.

D. Applying optimal schedules of representative states

The output, also called *policies*, of the SSADP planning approach, is a collection of system configurations annotated by time steps. When a decision is needed at a time step, with the current system state, one can choose and perform the policy of its most similar state in the collection, or derive one action by means of Equation III.3. The policies selected this way are conditioned upon the realizations of the demands and generation outputs in the components of s_t^{β} , so they represent a conditional scheduler.

$$\pi(s_t^{\beta}) = \operatorname{argmin}_{u_{t+1}^{\beta'}} \{R(s_t^{\beta}, u_{t+1}^{\beta'}) + \sum_{s'} Pr(s'|s_t^{\beta}, u_{t+1}^{\beta'}) V_{t+1}(s') + \alpha G(s_t^{\beta}, u_{t+1}^{\beta'})\} \quad (\text{III.3})$$

In addition to computational merit, the SSADP approach has a significant advantage over the DSADP method – it does not require knowledge of the initial system state in the planning phase, while the DSADP method does. The reason is that the DSADP method relies on the initial state to find system states and evaluate them. In SSADP, when its planning phase is completed, the resultant policies can be used for any initial status. Given a state with commitment schedules, operating

times, renewable generations, and the user demand profiles, the SSADP method can use Equation III.3 to determine a decision.

IV. EXPERIMENTAL RESULTS

We experimented with the proposed method on two variants of the test problem adopted from [4]. Our results showed that the SSADP solutions provide a better balance between generation cost and risk of failure to meet demand than deterministic methods. For large problems, the SSADP method is more efficient than the DSADP method.

A. Experimental Conditions

The variants extended the original problem in [4] with uncertainties in the demand. No non-dispatchable generators were used, so the net demand is equal to the total demand. The generation cost of a committed unit is computed as a quadratic function of the produced amount of power by the unit. We assumed that the demand follows a Gaussian distribution ($D_t \sim N(\bar{D}_t, \sigma_t^2)$) with a standard deviation of 5% of the expected demand at any time step, and it follows an AR(1) procedure over time steps. 1000 MCMC demand samples are generated and used for assessing the quality of the schedules from the various algorithms. The risk compensation cost $g_t(u_t, d_t)$ is given by $\alpha \cdot C_{FSO} \cdot \int_{\sum_i u_i^{cap_i}}^{\infty} \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left(-\frac{(D-\bar{D}_t)^2}{2\sigma_t^2}\right) \cdot dD$ where α is the proportionality constant, C_{FSO} is the full system operating costs (the cost of the system in which all units are turned on and generate according to their maximum capacity), and the integral is the failure probability (risk).

The SSADP method was implemented and compared against three existing algorithms: the priority list method [1], the unit decommitment algorithm [4], and the non-deterministic DSADP algorithm [3]. To decrease the risk level, we increase α in Equation II.2 for SSADP and increase the target reserve level for deterministic approaches. The experiments were performed on a desktop computer with Intel Core 2 Duo E6600 CPU (2.40GHz) in MATLAB 7.9.0.

B. 20-unit example

In this experiment we used 20 generators for 24-hour planning [4]. Figure IV.1 shows the performances of the priority list, the unit decommitment, the DSADP, and the SSADP methods in terms of the blackout risk and the effective extra cost. In the SSADP, the target reserve β starts from -50% to 80% with an increment 20%. The y-axis shows the risk that the system fails to meet the demand, whereas the x-axis shows the effective extra cost in percentage, which is calculated in three steps for each of the implemented algorithms. First, for each demand trial, the algorithm calculates the cost of meeting

the demand using their schedules obtained from the planning phase. Second, the costs across the demand trials are averaged. Third, the average cost translates into a percentage as a ratio of its difference from a *reference cost* divided by the reference cost. In this example, the reference cost refers to the point in the SSADP curve of the figure where the effective extra cost is 0.0 (because it is relative to itself). The reference cost 1207519.31 is obtained when the risk coefficient α is 1.0. It can be seen that the SSADP method achieves a better cost/risk balance than the DSADP method – for the same risk level, the SSADP costs less; for the same cost level, the SSADP is more reliable. Further, the SSADP method saves costs by several percents from the schedules of the priority list and the unit decommitment method at comparable risk levels.

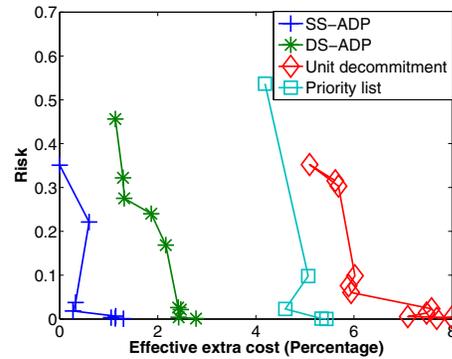


Figure IV.1. Performances of the algorithms on a 20-unit problem

We analyzed how the Maximum Committed Capacities (MCCs) of generated configurations distribute in the demand space and presented the results in Figure IV.2. In the figures, the x-direction shows the time steps, while the y-direction shows the target demands, the MCCs of the states, and the system MCC (the sum of the maximum capacities of all generators). The data shows that the states identified by the QP method covers the demand space in a fine granularity. This may demonstrate the effectiveness of the SSADP method in finding representative states.

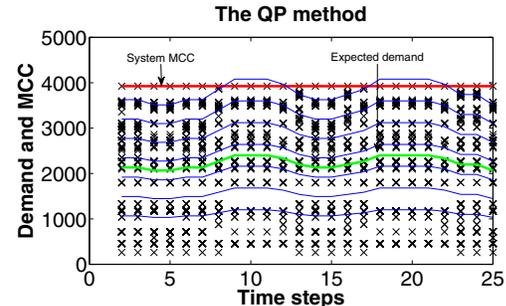


Figure IV.2. SSADP state space approximation where lines are for selected demands while crosses for MCCs of generated states

In terms of computational efficiency, to construct policies

for different risk-related penalty coefficients α , on the average, the priority list, the unit decommitment, SSADP and DSADP method respectively took 0.5, 47.7, 1822.31, and 2763.1 CPU seconds. Hence, the non-deterministic method took longer than the deterministic method. Meanwhile, the SSADP method is faster than the DSADP method, but due to the fact that this is not a large UC problem, the difference is not significant.

C. 40-unit example

Our next test example has 40 units for a weekly planning (168 hours). To create the example, we duplicate the 20 units and repeat the 24-hour demand profile seven times. Figure IV.3 shows the performances of the SSADP, priority list, and the unit decommitment methods. The curve for the DSADP is missing because it does not complete plan construction after a reasonably long period of time (20 hours). The conclusion is similar – the SSADP method outperforms the deterministic approaches in providing a better balance between the operating costs and the blackout risks. In terms of computational efficiency, to construct policies for different risk-related penalty coefficients α , on the average, the priority list, the unit decommitment and the SSADP method respectively took 6.5, 2993.5, and 3947.9 seconds. The unit decommitment methods took much longer than the priority list method because it has to carry out dynamic programming of 168 steps to find an optimal configuration for one generator at each of its decommitment iterations. It is also noted that the SSADP method is significantly faster than the DSADP method (more than 20 hours) that suffers from the potentially exponential growth problem.

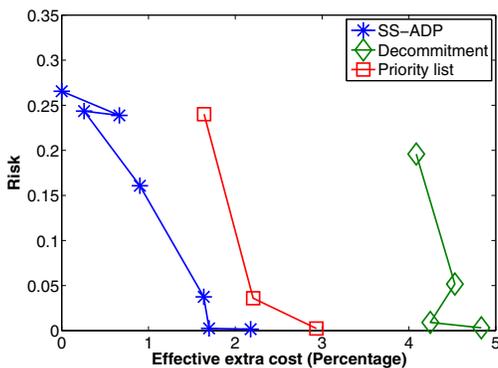


Figure IV.3. The performances of the algorithms on a 40-unit problem

V. CONCLUSIONS AND FUTURE WORK

We have proposed an approximate dynamic programming method to solve stochastic unit commitment problems where uncertainties exist in the power demands and the generation outputs of intermittent renewable energy sources. The method works with a reduced state space whose size is polynomial

in the number of generators and the number of selected target demand levels. We also proposed a functional metric to measure the similarity of the states in the reduced space. We proposed an approximate dynamic programming method in which when a state's value is not updated, its most similar state can be found and its value can be used instead. The proposed method yields lower costs and lower operational risk than the deterministic methods, and can solve larger problems than a previously developed decision space approximate method that relies on a tree structure and suffers from an exponential growth problem.

In the future, the SSADP approach can be improved in a number of ways. First, to improve the solution quality, a more sophisticated functional metric is possible. The current metric design does not consider the operational time that is a component of a system state. Incorporating it to a metric would enable the metric to account for the cost difference resulting from the operational times. This is especially true when the startup or shutdown costs of a generator depend on their operational durations. Second, to improve the efficiency of SSADP, a clustering technique can be used on top of the proposed metric. The idea is to build the clusters from the generated representative system configurations. Intuitively a cluster is a collection of system states within which the similarity of any two states is smaller than a threshold. In value updates for a given state, if its successor states do not belong to the reduced state space, the procedure of searching for the most similar state can be performed at the cluster level rather than at the state level. Therefore, the complexity reduces to the number of the clusters. It should greatly reduce the search time and thus accelerate the SSADP method.

REFERENCES

- [1] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*. New York: Wiley-Interscience, 1996.
- [2] S. Takriti, J. R. Birge, and E. Long, "A stochastic model of the unit commitment problem," *IEEE Transactions on Power Systems*, pp. 1497–1508, 1996.
- [3] D. Nikovski and W. Zhang, "Factored markov decision process models for stochastic unit commitment," in *IEEE Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply*, September 2010.
- [4] C. Li, R. B. Johnson, and A. J. Svoboda, "A new unit commitment method," *IEEE Transactions on Power Systems*, pp. 113–119, 1997.
- [5] M. L. Puterman, *Markov Decision Processes—Discrete Stochastic Dynamic Programming*. New York, NY: John Wiley & Sons, Inc., 1994.
- [6] C. Boutilier, R. Dearden, and M. Goldszmidt, "Stochastic dynamic programming with factored representations," *Artificial Intelligence*, pp. 49–107, 2000.
- [7] A. Martelli and U. Montanari, "Additive and/or graphs," in *the Third International Joint Conference on Artificial intelligence*, 1973, pp. 1–11.