MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Operational Planning of Thermal Generators with Factored Markov Decision Process Models

Nikovski, D.

TR2013-048 June 2013

Abstract

We describe a method for creating conditional plans for controllable thermal power generators operating together with uncontrollable renewable power generators, under significant uncertainty in demand and output. The resulting stochastic sequential decision problem has mixed discrete and continuous state variables and dynamics, and we propose a discretization method for the continuous part of the model that unifies all variables into a large discrete Markov decision process model. Although this model is way too large to be solved directly, its state transition probabilities can be factored efficiently, and a reduction of all continuous variables to one net demand variable makes it tractable by dynamic programming over a suitably constructed AND/OR tree. The proposed algorithm outperformed existing non-stochastic solvers on several problem instances, resulting in both lower risks and operational costs.

International Conference on Automated Planning and Scheduling (ICAPS)

Operational Planning of Thermal Generators with Factored Markov Decision Process Models

nikovski@merl.com
Mitsubishi Electric Research Laboratories,
201 Broadway, Cambridge, MA 02139, USA

Abstract

We describe a method for creating conditional plans for controllable thermal power generators operating together with uncontrollable renewable power generators, under significant uncertainty in demand and output. The resulting stochastic sequential decision problem has mixed discrete and continuous state variables and dynamics, and we propose a discretization method for the continuous part of the model that unifies all variables into a large discrete Markov decision process model. Although this model is way too large to be solved directly, its state transition probabilities can be factored efficiently, and a reduction of all continuous variables to one net demand variable makes it tractable by dynamic programming over a suitably constructed AND/OR tree. The proposed algorithm outperformed existing non-stochastic solvers on several problem instances, resulting in both lower risks and operational costs.

Introduction

The operational planning of thermal generators is a difficult sequential optimization problem that electrical power utilities must solve continuously to ensure that they meet power demand with maximal reliability and at a minimal cost. Fossil-burned thermal generators (using coal, natural gas, or oil) consume vast amounts of expensive fuel and contribute significantly to global warming, so minimizing the amount of consumed fuel is of primary importance in the electrical power industry.

Given a set of generators with their cost structure and fuel consumption rates, the objective of optimal operational planning is to find the best sequence of commands to turn individual generators on or off, and the optimal amount of power produced by each of them over an extended period of time, subject to the operational constraints that these generators might have. Typical planning periods range between one day or one week, and the state of the generators can typically be changes once every hour. The predicted demand over the entire planning horizon is also assumed to be known, either exactly, or with some quantifiable uncertainty.

There are several reasons why this problem is very computationally challenging. The first reason lies in the temporal constraints on the operational durations of individual generators that arise from their mechanical construction and requirements for reliable operation. It is generally not desirable (or frequently even possible) to turn the burners of the generators on and off at arbitrary moments, because frequent switching would cause damage due to excessive thermal expansion and contraction. For this reason, once a generator is turned on or off, it must remain in that state for several hours, and conversely, if it has been turned off, it must be kept off for several hours. In other words, when a command is given to turn a generator on or off, it is committed to that state for multiple decision periods, and that is why this problem is also known as the *unit commitment* problem. Due to these temporal constraints, the planning problem must be solved over the entire planning horizon, making it a sequential decision problem. Although the states of the generators are Boolean variables (on or off), the state variables of the sequential decision problem must include information about how many decision periods the generator has been on or off, and that increases the cardinality of the state space enormously.

The second reason for the high computational complexity of this problem is its mixed continuous/discrete nature. Some of the decision variables are discrete (e.g. the commitment status of generation units), and others are continuous (e.g. the amount of power produced by each unit). Moreover, the dynamics that govern the evolution of the system are also mixed: the total demand for electricity is a continuous scalar variable, whereas the main components (generation units) switch between discrete modes (on or off). This significantly limits the number of solution methods that can be applied to this problem, because there are relatively few planning and optimization methods that can solve mixed continuous/discrete problems efficiently.

The third reason this problem is very difficult is that at least part of the system dynamics are random, for most practical situations. In all cases, for any future moment during the planning period, the total demand for power will not be a completely known, deterministic value, but a random variable predicted from the information available at the time of planning. The prediction error can often be quantified (typical values are around 2% of total demand), and although most currently deployed planning systems have chosen to ignore this uncertainty or deal with it in a heuristic manner, such uncertainty information can arguably be used to improve the performance of the planning algorithm. Moreover, the increased penetration of renewable power sources such as photovoltaic panels and wind turbines, whose output depends strongly on uncontrollable atmospheric conditions such as solar radiation and wind speed, has effectively introduced much higher levels of uncertainty in the net demand for power to the controllable thermal generators. For example, if 20% of the power generator by a utility is supplied by wind turbines, in case the wind dies down suddenly, the net demand to the thermal generators might increase suddenly by 20%. The operational plan must allow for such contingencies, if forced outages are to be avoided. As a result, ignoring uncertainty in the system is becoming increasingly impossible for electrical power utilities. And, in addition, other sources of uncertainty are possible faults in the generators, which are naturally random events, but can be characterized probabilistically.

Due to its primary economic significance, the operational planning problem for thermal generators has been addressed by a very large number of solution methods, including ones based on dynamic programming, Lagrangian relaxation, interior point methods, and mixed integer programming, as well as heuristic methods such as genetic algorithms, simulated annealing, evolutionary programming, differential evolution, particle swarm optimization, Hopfield neural networks, etc. (Wood and Wollenberg 1996; Xia and Elaiw 2010). Formulations as a model predictive control or optimal control problem are possible, too (Xia, Zhang, and Elaiw 2011). Dynamic programming methods can leverage successfully the sequential nature of the decision process in order to compute suitable plans efficiently, but suffer from the well known curse of dimensionality due to the large size of the state space of the problem. Mixed integer programming methods can handle successfully the mixed continuous/discrete nature of the planning task, but again do not scale up very well because of the sheer combinatorial complexity of the discrete optimization part. Lagrangian relaxation could also be a very effective solution to the mixed continuous/discrete optimization problem, and has been shown to perform well on large problems. Global optimization methods such as genetic algorithms and simulated annealing can be very effective on problems with disjoint feasibility regions, but cannot guarantee that global optima would always be reached, in general.

However, the majority of these methods either ignore uncertainty completely at the planning state, or deal with it heuristically, or consider only a small number of possible future realizations of the uncertain variables, and usually compute a fixed operational plan for the entire period that is executed sequentially. As is well known in AI, such plans can only succeed if the problem domain is static, completely observable, deterministic, and the action descriptions available to the planner are correct and complete. One common heuristic is to include a safety margin of extra capacity (for example, 3%) to be committed for production. This results in operating more and/or larger units than are necessary to meet expected demand. This approach is largely heuristic, and is not likely to work in the future, when renewable energy sources become even more widespread. And, in general, whereas there might be some value in algorithms that can find fixed plans that are maximally robust with respect to future uncertain outcomes, a much more natural approach would be to use algorithms that can compute conditional plans that can select actions depending on future states (also known as contingency plans in AI, feedback controllers in control theory, and decision policies in operations research). This paper describes one such approach based on factored Markov decision processes (fMDP), where continuous dynamics are discretized by means of a barycentric approximation and added to the discrete dynamics, the state of the resulting completely discrete fMDP is pruned by means of problem domain knowledge, and the optimal decision policy is found by means of dynamic programming over AND/OR trees.

Formulation of the Planning Problem

Formally, the operational planning problem for generators can be described as follows. Given N available controllable generator units, and a planning horizon of length T units of suitable duration, for example one hour, the overall goal is to minimize the total operating cost for these units, subject to operating constraints and at an acceptable risk of a forced outage. The demands D_t, $1 \leq t \leq T$, over the entire planning horizon are random variables coming from a stochastic process with known structure and parameters. There are also K uncontrollable generators, and we assume that the realizations y_t^k of their random output amounts Y_t^k, $1 \leq t \leq T$, $1 \leq k \leq K$, also come from known stochastic processes. At all times, the sum of the supply from all generators, controllable and uncontrollable, must match the total demand at that time.

In order to formulate a sequential decision problem, we introduce the decision variables $u_t^i \in \{0, 1\}$ for all time periods t, $1 \leq t \leq T$, and controllable units i, $1 \leq k \leq K$, which represent the intended commitment status of the generators during the next operational period. Similarly, we introduce the state variables $x_t^i \in \{-l, -l+1, \ldots, -1, 1, \ldots, L-1, L\}$, where l is the minimum allowed time for keeping a generator off, and L is the minimum allowed time for keeping a generator on. Negative values correspond to off condition, and positive values correspond to on condition.

For the state variables of the controllable part of the process, if we have an existing commitment status u_{t-1}^i for generator i, operation time x_{t-1}^i, and new commitment status u_t^i, the new operational time x_t^i can calculated by Equation (1) where T_i^{cl} is the "cold start" time of unit i, l_i is the minimum down time of unit i, and L_i is the minimum up time of unit i (Li, Johnson, and Svoboda 1997).

$$x_t^i = \begin{cases} 1 & \text{if } -T_t^{cl} \le x_{t-1}^i \le -l_i \text{ and} \\ & u_t^i = 1 \text{ (start up)} \\ x_{t-1}^i + 1 & \text{if } 1 \le x_{t-1}^i \le L_i - 1 \\ & \text{(up and must stay up)} \\ L_i & \text{if } x_{t-1}^i = L_i \text{ and } u_t^i = 1 \\ & \text{(up and available to shut down)} \\ -1 & \text{if } x_{t-1}^i = L_i \text{ and } u_t^i = 0 \\ & \text{(shutting down)} \\ x_{t-1}^i - 1 & \text{if } -l + 1 \le x_{t-1}^i \le -1 \\ & \text{(down and must stay down)} \\ & \text{or } -T_i^{cl} + 1 \le x_{t-1}^i \le -l_i \text{ and} \\ & u_t^i = 0 \\ & \text{(down and available to start up)} \\ -T_i^{cl} & \text{if } x_{t-1}^i = -T_i^{cl} \text{ and } u_t^i = 0 \end{cases} \quad (1)$$

Additional constraints, such as maximal up/down times, can be accommodated by suitable modifications to Eq. (1).

For the demand variable, we assume that we have a stochastic dynamic model that specifies the probability $Pr(D_t = d_t | D_{t-1} = d_{t-1}, D_{t-2} = d_{t-2}, \dots, D_0 = d_0)$ that value (power demand) d_t will be observed at time t if a series of demands $d_0, d_1, \dots, d_{t-2}, d_{t-1}$ has been observed until then. Similarly, for each uncontrollable generator k we assume that we can estimate the probability $Pr(Y_t^k = y_t^k | Y_{t-1}^k = y_{t-1}^k, Y_{t-2}^k = y_{t-2}^k, \dots, Y_0^k = y_0^k)$ that value (power output) y_t^k will be observed at time t if a series of outputs $y_0^k, y_1^k, \dots, y_{t-2}^k, y_{t-1}^k$ has been observed until then. Various predictive models can be used, such as auto-regressive (AR), neural nets, support vector machines, etc., that map past observations onto future values.

The planner must observe several constraints in minimizing the total cost. The load balance constraint states that the total generation must be equal to the demand d_t at any time step. If p_t^i is the generation of unit i at hour t, then

$$\sum_{i=1}^{N} p_t^i u_t^i + \sum_{k=1}^{K} y_t^k - d_t = 0, \text{ for } \quad t = 1, 2, \dots, T. \quad (2)$$

The objective function is presented in Equation 3, where $\mathbf{E}_{u_0,x_0,y_0,d_0}$ 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. For notational simplicity, the decision variables at time t are represented as the vector $u_t \doteq [u_t^1, u_t^2, \dots, u_t^N]$, the state variables are denoted by the vector $x_t \doteq [x_t^1, x_t^2, \dots, x_t^N]$, and the realizations of all uncontrollable generators are denoted as $y_t \doteq [y_t^1, y_t^2, \dots, y_t^K]$.

$$\begin{aligned} J^* \;=\; & \min_{u_1,u_2,\dots,u_T} \mathbf{E}_{u_0,x_0,y_t,d_t} \{ \textstyle\sum_{t=0}^{T-1} [\sum_{i=1}^{N} f_i(x_t^i, u_t^i, y_t, d_t) \\ + \; & \textstyle\sum_{i=1}^{N} h_i(x_t^i, u_t^i, u_{t+1}^i) + g_t(u_t, y_t, d_t)] \} \end{aligned} \quad (3)$$

Here $f_i(x_t^i, u_t^i, y_t, d_t)$ denotes the operating cost of operating unit i in configuration u_t^i and state x_t^i for one time step in order to meet demand d_t when the uncontrollable generators output electricity amount y_t. The function $h_i(x_t^i, u_t^i, u_{t+1}^i)$ denotes the cost of switching to configuration u_{t+1}^i at the end of the step. The third cost component, $g_t(u_t, y_t, d_t)$, denotes the equivalent cost of the risk of not being able to meet demand d_t under output of uncontrollable generators y_t with the chosen configuration of all units u_t. This cost is proportional to the probability that the total capacity of the committed units in u_t plus what the uncontrollable generators produce (y_t) is less than the demand d_t:

$$g_t(u_t, y_t, d_t) = \alpha Pr(\sum_{i=1}^{N} u_t^i cap^i + \sum_{k=1}^{K} y_t^k < d_t),$$

where cap^i is the maximal generation capacity of unit i. A suitably chosen proportionality coefficient α specifies the relative preference between minimizing operating cost and risk of failure to meet demand. By adding the operating cost and risk compensation cost together, the objective function represents a trade-off between fuel costs and risk.

At any given time, if we can find the optimal sequence $u_1, u_2, \dots, u_T$ that minimizes the cost in Equation 3 by whatever computational means, we will have an operational plan that can be executed over the entire planning horizon. However, as argued above, such an open-loop, unconditional plan is not tailored to the concrete situation that will be encountered in the future. An alternative approach is to recognize that the uncertainty in power demand and generator supply makes the decision problem a stochastic one, and its optimal solution is not an unconditional plan (sequence of commitment decisions), but an entire decision policy. A conditional operational planner could compute conditional plans that are robust to future variations of supply and demand, and could provide a safety margin implicitly, by planning for all possible contingencies. One significant difficulty associated with this approach has been how to represent all such possible contingencies, and how to plan for them. One proposal organizes all future possible realizations of the system (called scenarios) as a tree of scenario bundles (Takriti, Birge, and Long 1996). However, this model for representing stochasticity is limited to only the few scenarios included in it, whereas in a practical system the future evolution can be realized in an infinite number of ways. Our work aims to expand this approach by improving the probabilistic modeling of system evolution.

We propose a method for finding the optimal conditional operational plan of a set of power generators under stochastic demand for electrical power and stochastic output of some generators. Unlike traditional operational plans, which are fixed in advance, a conditional operational plan depends on the future state of the observable random variables (demand and output), and can result in different actual sequences of decisions depending on the observed outcomes for these variables. The planner explicitly balances the operational cost of electricity generation with the risk of not being able to meet future electricity demand. We represent the stochastic dynamics of the components of the system as a factored Markov decision process (MDP) model, and propose efficient approximate algorithms for computing suitable conditional operational plans.

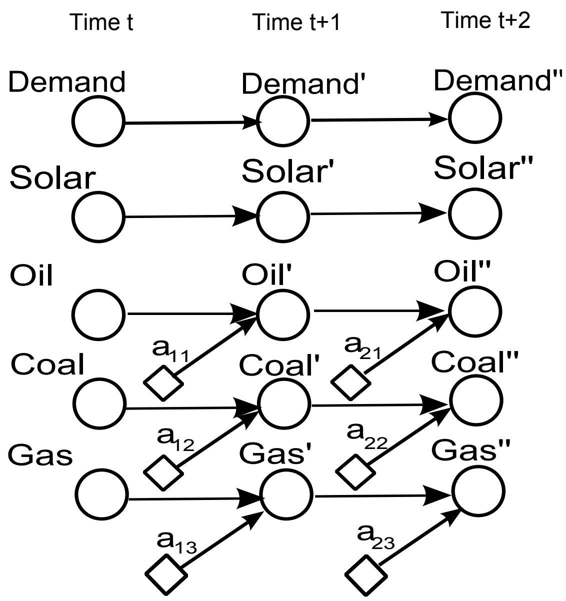

Figure 1: DBN for a power generation problem with three controllable and one uncontrollable power generators.

Factored Markov Decision Processes for Conditional Operational Planning

We propose to represent a power generation system consisting of multiple generators of the type described above by means of a *factored Markov decision process* (fMDP), and find the optimal conditional operational plan by means of approximate dynamic programming (Boutilier, Dearden, and Goldszmidt 2000). The fMDP is usually expressed graphically as a *dynamic Bayesian network* (DBN). A DBN consists of circles that represent random variables, diamonds that represent decision variables, and directed edges connecting the circles and diamonds that represent the statistical dependence between the corresponding variables. When dealing with a time-dependent system, each time period (e.g. one hour) is represented by its own set of random variables. Three time slices of the DBN for an example stochastic unit commitment problem with four generators, one of which uncontrollable (solar), are shown in Fig. 1.

In Fig. 1, one random, continuous, and uncontrollable variable represents the power demand. Another random, continuous, and uncontrollable variable represents the output of a photovoltaic generator. (In this case, these two model components are first-order Markovian, that is, the next state depends only on the current state, for example by means of an AT(1) model. However, this is not a fundamental limitation: for higher-order models, edges from previous time slices can be added, too.) In addition, three conventional controllable generators are shown, too; their discrete variables x_t^i take on $l + L$ possible different values, and represent the operational time of the respective generator. Three decision variables (shown as diamonds) represent the individual decisions $a_{ti} = u_t^i$ to turn on/off the corresponding generators, and thus commit them for power production. These models components are necessarily first-order Markovian, but do not need to be deterministic — certain probability of failure to change the state of a generator as desired could be modeled in them. The probabilistic evolution of the system is described by local conditional probability tables for each variable, where the conditional dependence is defined only on the parents of that variable in the graph of the DBN. Thus, the DBN serves as a compact representation of a large Markov decision process whose state space is exponentially large in the number of states of the individual variables over which it is factored.

In order to specify a factored MDP, the state, action, and transition model for each individual variable must be defined, along with the reward/cost structure. This is done differently for the thermal generators which are naturally represented by means of discrete variables, and for the demand and uncontrollable generators which are naturally represented by means of continuous variables. For the fMDP part corresponding to thermal generators, the definitions of state and action variables coincide with those in the original sequential decision problem described in Section . For the continuous variables, we use a discretization method based on barycentric coordinates that we have already applied to other sequential decision problems such as train run-curve optimization and set-point scheduling for air conditioners (Nikovski and Esenther 2011; Nikovski et al. 2012; Nikovski, Xu, and Nonaka 2013).

The main idea of the method is to replace the continuous state variables with a discrete set of states in a way that approximates well the original continuous dynamics. Let the dynamics of a continuous component of the model be represented by the function $z_{t+1} = f_z(z_t, a_t)$, where z_t is a vector variable that could include one or more of the demand d_t, the output of uncontrollable generators y_t^k, or some of their time-lagged values d_{t-1}, y_{t-1}^k, etc. Let the dimensionality of this vector be b. The objective of the conversion method is to represent the dynamics of the continuous system $z_{t+1} = f_z(z_t, a_t)$ by a conditional probability transfer function $Pr(s_{t+1} = s^{(j)} | s_t = s^{(i)}, a_t = a^{(k)})$, defined over suitably chosen set S of N discrete states $s^{(i)}$, $1 \leq k \leq K$. The algorithm selects N states $s^{(1)}$, $s^{(2)}$, ..., $s^{(N)}$ such that each corresponds to a state $z^{(i)} \in R^b$, and their Delaunay triangulation is computed (Fig. 2), (Preparata and Shamos 1990). Then, each available action $a^{(l)}$ is executed in each of them in turn, according the continuous dynamics function $z' = f_z(z^{(i)}, a^{(l)})$, and the barycentric coordinates $p_1, p_2, \ldots, p_{b+1}$ of the end state z' are computed with respect to the simplex that encloses it. These barycentric coordinates are then used as transition probabilities of the discrete MDP. The detailed computational procedure, along with discussion of its computational complexity, is available in (Nikovski and Esenther 2011).

Conceptually, we can think of this algorithm as a way of converting the system dynamics represented by the function f_z to an equivalent probabilistic representation involving only a small set of points $s^{(i)}$embedded into the original continuous state space of the system. If the system starts in one of these few points, the successor state z', in general, will

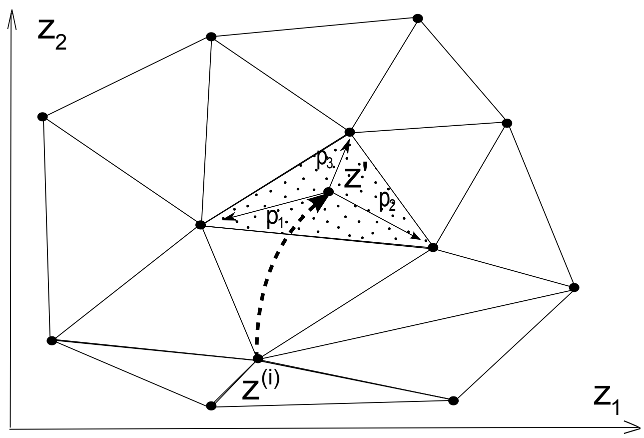

Figure 2: A Delaunay triangulation on a set of vertices sampled from the embedding two dimensional space. The dashed line shows the transition from some starting state $z^{(i)}$ under action a resulting in end state $z' = f(z^{(i)}, a)$. The simplex (here, triangle) containing the end state z' is shown with a dotted background, and the barycentric coordinates p_1, p_2, and p_3 of z' are computed with respect to the vertices of that simplex. These coordinates are also the transition probabilities from $z^{(i)}$ under action a to the states corresponding to these vertices in the resulting MDP.

not coincide with another one of these points. However, we can identify the $b+1$ points that define a simplex that completely encloses the successor state z', and can think that the system has transitioned not to point z' itself, but to the vertices of this simplex with various probabilities, instead. The probabilities are equal to the convex decomposition of point z' with respect to the vertices of the simplex, also known as the barycentric coordinates of that point within the simplex. The similarities between convex combinations (barycentric coordinates) and probability mass functions required by the MDP formalism make this conversion possible.

This procedure is applied in turn for every group of variables in the DBN that have temporal dependence. For the demand variable D, we could assume that the next demand D_{t+1} depends only on the current demand D_t (Markovian property of the underlying stochastic process) with transition probability $Pr(D_{t+1} = d_{t+1}|D_t = d_t)$, and if a higher-order model is necessary, time-lagged values of demand D_{t-1}, D_{t-2}, etc. could be included. For the uncontrollable generators, we make similar assumptions that Y^k_{t+1} depends only on Y^k_t, with probability $Pr(Y^k_{t+1} = y^k_{t+1}|Y^k_t = y^k_t)$. These transition probabilities can be estimated either from statistical data, or by means of discretizing a suitable continuous stochastic Markov process, such as the auto-regressive process of order 1 (AR(1) process).

Once the transition probabilities for all variables in the DBN have been determined, joint transition probability for the entire system $Pr(u_{t+1}, x_{t+1}, y_{t+1}, d_{t+1}|u_t, x_t, y_t, d_t)$ can be computed from the transition probabilities of the individual random variables, as is customary for Bayesian networks. It can be observed that although the MDP has a very large joint state space, its transition structure is very sparse.

The next step is to determine the transition cost, which, unlike transition probabilities that can be specified separately for each individual variable, must be specified for the entire MDP. Given a joint 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 computed as

$$\begin{aligned} c(u_t, x_t, u_{t+1}, y_t, d_t) &= \textstyle\sum_{i=1}^N f_i(x^i_t, u^i_t, y_t, d_t) \\ + \textstyle\sum_{i=1}^N h_i(x^i_t, u^i_t, u^i_{t+1}) &+ g_t(u_t, y_t, d_t) \end{aligned} \tag{4}$$

where the switching costs $h_i(x^i_t, u^i_t, u^i_{t+1})$ and risk cost $g_t(u_t, y_t, d_t)$ are computed as described above, and the fuel costs $f_i(x^i_t, u^i_t, y_t, d_t)$ are computed by solving the following economic dispatch problem: minimize $\sum_i F_i(p^i_t)$ subject to the generation limits for all generators and the load balance constraint for this particular realization of the uncontrollable variables y_t and demand d_t:

$$\sum_{i=1}^{N} u^i_t p^i_t + \sum_{k=1}^{K} y^k_t - d_t = 0$$

where $F_i(p^i_t)$ is the cost of producing p^i_t units of electricity by generator i; typically, this function is quadratic in p^i_t, and the economic dispatch problem can be solved by means of quadratic programming. The objective of economic dispatch is to find the optimal generation amounts p^i_t of the committed units so that the cost of generation is minimized for a specific realization of the random variables. After the optimal generation amounts $[p^1_t, p^2_t, \ldots, p^N_t]$ are found, the individual generation costs can be calculated as $f_i(x^i_t, u^i_t, y_t, d_t) = F_i(p^i_t)$, $1 \le i \le N$.

Given such an MDP, we can define its cost-to-go functions J_t for each step t and each joint state of the MDP. For the terminal step T, when no further decisions will be made, $J_T(u_T, x_T, y_T, d_T) = 0$.

For all other steps, the cost-to-go function $J_t(u_t, x_t, y_t, d_t)$ is defined iteratively by means of a Bellman equation, as follows (Puterman 1994):

$$\begin{aligned} &J_t(u_t, x_t, y_t, d_t) = \min_{u_{t+1}} \{c(u_t, x_t, u_{t+1}, y_t, d_t) \\ &+ \textstyle\sum_{d_{t+1}, y_{t+1}} Pr(d_{t+1}, y_{t+1}|d_t, y_t) J_{t+1}(u_{t+1}, x_{t+1}, y_{t+1}, d_{t+1})\} \end{aligned} \tag{5}$$

Note that the transition probabilities $Pr(d_{t+1}, y_{t+1}|d_t, y_t)$ are factored conveniently, due to the conditional independence relations in the DBN of the MDP. The cost-to-go function $J_0(u_0, x_0, y_0, d_0)$ of the initial state of the generators and demand would then correspond to the minimal operating cost under the optimal policy for the entire planning problem.

In principle, this cost can be found by computing the costs-to-go of all states in the MDP. However, when some of the variables are continuous, the cost-to-go (value function) of the MDP cannot be computed and represented efficiently. The discretization method described above addresses precisely this problem, by replacing the continuous variables D and Y_k with sets of discrete states S, making the entire

MDP discrete, and standard MDP solution methods such as dynamic programming, value iteration, and policy iteration can be applied (Puterman 1994). A solution method based on dynamic programming over AND/OR trees is described in the next section.

Furthermore, if these costs are computed and stored, the optimal decision $u_{t+1} = \pi_t(u_t, x_t, y_t, d_t)$ for time step t and state (u_t, x_t, y_t, d_t) can be identified as the one that minimizes the right-hand side of the Bellman equation 5:

$$\pi_t(u_t, x_t, y_t, d_t) = argmin_{u_{t+1}}\{c(u_t, x_t, u_{t+1}, y_t, d_t) + \sum_{d_{t+1}, y_{t+1}} Pr(d_{t+1}, y_{t+1}|d_t, y_t) J_{t+1}(u_{t+1}, x_{t+1}, y_{t+1}, d_{t+1})\} \quad (6)$$

This policy is conditioned upon the current realizations of the random variables y_t and d_t, so it represents a conditional planner. By observing the outcomes y_t and d_t for each consecutive time step, different actual operating schedules will be obtained.

Solving fMDP Models with Aggregated Net Demand

The objective of solving the stochastic unit commitment problem represented by the fMDP is to find the optimal policy that maps the states of the fMDP onto the decision variables that signify which generators will be turned on/off in the next period, where optimality is defined in terms of jointly minimizing production cost and risk of failure. The straightforward method of solving fMDPs is to expand the factored state and solve the resulting flat MDP by means of dynamic programming, applying equation 5 repeatedly, starting from the terminal step and proceeding backwards to the first step (Puterman 1994). However, for most practical problems, e.g. when $L = l = 5$, the number of generators $N = 20$, the number of one-hour time periods $T = 24$, the expanded MDP will have $|X| = T(L+l)^N = 24 \cdot 10^{20}$ distinct states for the controllable generators only, and would be impossible to solve.

One practical simplification of the problem is to aggregate the output of the uncontrollable generators Y_t into the demand variable, by subtracting these outputs from the total demand D_t to arrive at the net demand D'_t. If all uncontrollable random variables are Gaussian processes, then D'_t is a Gaussian process, too, with expected value (mean) $\bar{D}'_t$ and variance σ_t for each time period t. Henceforth, we will assume that D_t denotes the net demand. For planning purposes, the net demand D_t can be computed by subtracting the expected values $\bar{Y}_t$ at the time of planning ($t = 0$). When executing the policy, the actually observed realizations y_t at time t can be used to estimate the distribution of the random variable D_{t+1}, so that the estimates of the transition probabilities $Pr(d_{t+1}|d_t, y_t)$ will in fact be based on y_t, when determining the optimal configuration u_{t+1} by means of Equation 6.

Another computational simplification of the problem is to reduce the size of the MDP in a reasonable manner. Intuitively, if forecasts for the values of the continuous random variables D_t and Y_t are known in advance, and the assumption that these are Gaussian processes holds true, most of the configurations of the generators u_t at time t would be irrelevant to satisfying demand at that time. Some of them will have capacities too low to meet demand, and others will use unnecessarily many generators to meet demand economically. By considering only configurations u_t of the controllable part of the MDP whose maximal committed capacity (MCC) is close to the expected net demand $\bar{D}_t$, we can drastically reduce the size of the space of the MDP.

A practical way of identifying such suitable configurations is to run a fast deterministic algorithm for unit commitment for several possible values of target reserve β such that the target demand is $(1+\beta)\bar{D}$. Suitable schedules S_β are identified for each β, and the generator configurations u_t present in S_β are included in the reduced state space of an approximate solver, which essentially switches between individual segments from multiple schedules S_β, depending on the time evolution of power demand and uncontrollable generators.

Hence, the fundamental idea of the solution algorithm is to identify suitable configurations for representative demands, and then use them to produce schedules for any possible realization of demand. We use an AND/OR tree (Martelli and Montanari 1973) to represent all selected configurations of the generators and possible realizations of future demand. The AND/OR tree is then used for planning for any demand instances.

The algorithm is outlined in Table 1. Before discussing its details, we remark that *commitment schedule* specifies whether a generation unit is on or off. A commitment schedule may specify the on/off status of the units over all time steps. When restricted to a particular step, it specifies the unit status at that step. For convenience, in the rest of the paper, a *schedule* refers to a commitment schedule unless stated otherwise.

1. Pick a set of demand instances
   and solve UCs to obtain schedules
2. Use the schedules to build an AND/OR tree
3. **Return** the AND/OR tree for planning

Table 1: The algorithm to solve the factored MDP

Generating Candidate Schedules

This step identifies a finite set of representative commitment schedules so that they can be reused in the remaining steps. To solicit schedules, we first select a set of demand samples in hope that they are representative ones. For each slected demand, a deterministic UC problem associated with the demand is solved to obtain its schedule. We start identifying demand samples by finding the overall "upper" and "lower" demands of interest. Let the mean of the demand D be $\bar{D} = [\bar{D_1}, \bar{D_1}, \ldots, \bar{D}_T]$. Starting from a large positive number β and decreasing it gradually, we find a demand $(1+\beta)\bar{D}$ whose UC is feasible. This demand is the upper demand. Similar procedure can find the lower demand. The two demands determine a demand interval. The schedule generation procedure performs search in the interval and finds demands and their solution schedules.

An iterative search procedure is as follows. Given a lower demand $\underline{d}$ and upper demand $\overline{d}$, a new demand $d(=(\overline{d}+\underline{d})/2)$ is created. (The demands $\underline{d}$ and $\overline{d}$ are called *parent demands*.) Its asscoiated UC problem $UC(d)$ is solved. If $UC(d)$ and its parent $UC(\underline{d})$ have different schedules, their average demand $(d+\underline{d})/2$ is added to consideration and the interval $[d, \overline{d}]$ is added to future search; otherwise, the interval $[d, \overline{d}]$ is discarded. By bookkeeping a demand and their parents, the procedure knows what interval it is searching. The advantage of this search strategy is that it focuses on the regions that lead to distinct schedules. This is contrast to an evenly split approach that searches an interval $[\underline{d}, \overline{d}]$ in a uniform manner. The uniformity in demand interval search does not necessarily mean schedule distinction.

The demands selected in iteratvie interval search preserve the *upward/downward* trend in the demand vector over time steps. A *upward* or *downward* trend is that the demand mean at the next step is larger or smaller than that at the current step. Since the lower and upper demands are proportional to the mean iterative interval search and the newly created demands are the arithmetic average of their parents demands, the trends are preserved. To diversify the set of the selected demands, we add some randomized schedules to the schedule set. Specifically, we randomly change the commitment schedules for a small portion of the identified schedules.

The pseudo-codes that implement those ideas are presented in Table 2. The constant `MinWidth` determinines the size/width of the intervals to be discarded. The other constant `MaxNum` is the maximum number of schedules to find. They are initialzied at Line 1. A queue data structure **Q** holds the processed and to-be-processed UCs. The pointer of the queue is Q_ptr. For a demand d, we use $UC(d)$ to refer to its associated UC problem and $u(d)$ to refer to the schedule of $UC(d)$. Line 2 initializes the queue and the schedule set **U** to be empty. Line 3 generates the "maximum" demand and adds the UC to the queue. Line 4 generates the "minimum" demand and adds the UC to the queue. Line 5 produces the first child demand and adds its UC to queue. It is pointed by the queue pointer at Line 6. Line 8 solves the UC. Line 9 loads the parent demands $\underline{d}$ and $\overline{d}$. Line 10 examines the schedule of $UC(d)$ and its "upper" parent $UC(\overline{d})$. If they are different, or they are the same but the interval is larger than the preset `MinWidth`, a new demand and its UC are created and added to the tail of the queue at Line 11. Lines 13-15 check the other parent likewise. Line 16 moves the pointer forward. Such a process terminates if the pointer points to `null` (no more UCs in the queue) or the maximum number of distinct schedules have been achieved (Line 7). Line 18 copies the set **U** with the identified schedules stored in the examined UCs of the queue. Line 19 perturbs the schedules by adding some randomization. Finally, Line 20 returns the schedules in **U**.

1. `MinWidth` ←constant, and `MaxNum` ←constant
2. $\mathbf{Q} \leftarrow \emptyset$, $\mathbf{U} \leftarrow \emptyset$
3. Find upper demand $\overline{d}$; Solve $UC(\overline{d})$; Add $UC(\overline{d})$ to **Q**
4. Find ower demand $\underline{d}$; Solve $UC(\underline{d})$; Add $UC(\underline{d})$ to **Q**
5. Add $UC(d)(d=(\overline{d}+\underline{d})/2)$ to **Q**
6. $Q\_ptr \leftarrow 3$
7. **While** $Q\_ptr \neq$ `null` && $Q\_ptr \leq$ `MaxNum`
8. Solve $UC(d)$ pointed by Q_ptr
9. $(\overline{d}, \underline{d})$← the parent demands of $UC(d)$
10. **If** $\texttt{differ}(u(d), u(\overline{d}))$ || $(\texttt{same}(u(d), u(\overline{d}))\&\&|d-\overline{d}| \geq \texttt{MinWidth})$
11. Add $UC(d)(d=(d+\overline{d})/2)$ to **Q**
12. **End** If
13. **If** $\texttt{differ}(u(d), u(\underline{d}))$ || $(\texttt{same}(u(d), u(\underline{d}))\&\&|d-\underline{d}| \geq \texttt{MinWidth})$
14. Add $UC(d)(d=(d+\underline{d})/2))$ to **Q**
15. **End** If
16. Q_ptr++
17. **End** of While
18. **U** ← UC schedules from Q_1 to $Q\_(ptr-1)$
19. Perturb **U**
20. **Return U**

Table 2: Generating schedules to be reused in the remaining algorithmic steps

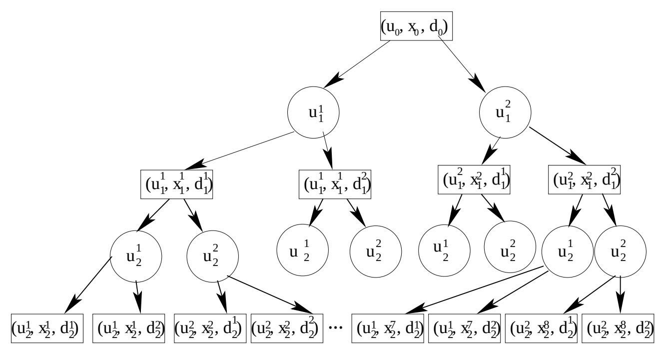

Figure 3: An AND/OR tree example

Building the AND/OR tree

An AND/OR tree has two types of nodes — AND nodes and OR nodes. An AND/OR tree is a tree where (1) its root is an AND node, (2) it has alternating levels of AND and OR nodes, and (3) its terminal nodes are AND nodes (Martelli and Montanari 1973). An AND/OR tree is shown in Fig. where the AND/OR nodes are respectively in rectangular/circular shapes. Note that in this case the outputs of the uncontrollable generators Y_t have been aggregated into the net demand variable D_t, and are not included in the AND/OR tree.

An AND node for the UC problem is associated with a system state (u_t, x_t, d_t) at time step t, whereas an OR node is associated with the action u_t at that time. The root node corresponds to the initial state of the UC system. The values of the nodes are evaluated bottom-up. For an OR node u_{t+1}, if its parent AND node is (u_t, x_t, d_t) and its children (AND) nodes are $\{(u_{t+1}, x_{t+1}, d_{t+1})|d_{t+1}\}$, then the value of the OR node is evaluated as

$$V_t(u_{t+1}|u_t, x_t, d_t) = c(u_t, x_t, u_{t+1}, d_t) + \sum_{d_{t+1}} p(d_{t+1}|d_t) V_{t+1}(u_{t+1}, x_{t+1}, d_{t+1}) \quad (7)$$

Note that the notation $V_t(u_{t+1}|u_t, x_t, d_t)$ means that the value of OR node is conditional on its parent AND node. For an AND node (u_t, x_t, d_t), its value $V_t(u_t, x_t, d_t)$ is evaluated as follows:

$$V_t(u_t, x_t, d_t) = \begin{cases} c(u_T, x_T, u_T, d_T), & \text{if } t = T \\ \min_{u_{t+1}} V_t(u_{t+1}|u_t, x_t, d_t) & \text{otherwise} \end{cases} \quad (8)$$

Note that the minimization in $\min_{u_t}$ is over all children OR nodes u_{t+1}, and that no configuration switching cost is incurred at the last step, since the continuation of the schedule at that time is yet unknown.

Now we are ready to explore the AND/OR tree for planning purpose. It takes two steps – building the AND/OR tree and constructing the schedules for planning.

1. The root node (u_0, x_0, d_0) is the initial state of the system. It is the first node of the tree. It is the node at Level 0. Since the levels correspond to the time steps in a UC, we use steps to refer to levels. The rest of the tree is built over time steps: the AND nodes at Step t and an OR node are used to build AND nodes at Step $t + 1$. The OR nodes are the schedules generated at Step 1 of the entire algorithm (Table 1). For a node (u_t, x_t, d_t), we use every generated schedule to produce AND nodes at the next time step. Let the OR node be u_{t+1}. The status of the units at next time step is (u_{t+1}, x_{t+1}). Since the demand at $t + 1$ is uncertain, we generate an AND node for every possible demand. So the set of next AND nodes is $\{(u_{t+1}, x_{t+1}, d_{t+1}|d_{t+1})\}$. The process repeats until completion at Step T. The pseudo-codes for this process are presented in Table 3. Let the notation $\mathbf{N}_t^A$ and be the AND nodes at Step t, and $\mathbf{N}_t^O$ be the schedules obtained from schedule generation but restricted to Step t.

Initialize the root node $\mathbf{N}_0^A$ to the initial system state
For $t = 1, \ldots, T$
  $\mathbf{N}_t^A \leftarrow \emptyset$
  $\mathbf{N}_t^O \leftarrow$ schedules from schedule generation
  For each node $(u_{t-1}, x_{t-1}, d_{t-1})$ in $\mathbf{N}_{t-1}^A$
    For each schedule u_t in $\mathbf{N}_t^O$
      For each demand d_t
        Node (u_t, x_t, d_t) is generated
        If the UC is feasible
          $\mathbf{N}_t^A \leftarrow \mathbf{N}_t^A \cup \{(u_t, x_t, d_t)\}$
        End if feasible
      End for each demand
    End for each schedule
  End for each node n
End for t
Return the tree represented by $\{\mathbf{N}_0^A, \mathbf{N}_1^A, \ldots, \mathbf{N}_T^A\}$.

Table 3: Building an AND/OR tree

2. The AND and OR nodes in the tree are evaluated by Equations (7) and (8). In evaluating the non-terminal AND nodes, there must be an OR node that achieves the minimum in Equation (8). The action represented by that OR node is the best action of the system state represented by the parent AND node.

Evaluating MDP Policies

Once a policy has been computed and stored in the AND/OR tree, we adopt a sampling approach to evaluate its operational cost and risk under future random demand D. For this purpose, we draw a suitable number of samples $d = [d_1, d_2, \ldots, d_T]$ from the demand variable D (e.g., 1000 samples). For each sample, we start from the root of the tree and execute the actions specified by the tree. Such an execution results in a path in the tree. Specifically, an execution path is a sequence of system states and actions $\{(u_0, x_0, d_0), u_1, (u_1, x_1, d_1), \ldots, u_T, (u_T, x_T, d_T)\}$ that are prescribed by the initial system state, the AND/OR tree, and the demand realization $d = [d_1, d_2, \ldots, d_T]$. The cost of a path can be accessed by solving the economic dispatch problem for each step, given the prescribed configurations u_t, while its risk can be calculated using the committed capacity u_t and the realization of demand d_t. The overall risks and costs are the average across the paths associated with the demand samples. These costs and risks show how the risks can be compromised by the additionally paid cost. Pseudo-code of the simulation procedure is presented in Table 4.

For each demand sample
  Determine the execution path from the AND/OR tree
  Solve the UC based on the path to get the cost
  Calculate the risk based on the path
  Sum up the cost
  Sum up the risk
End For each demand
Calculate the average cost and average risk
Return the average costs and risks

Table 4: Simulating the cost and risk of an MDP policy

Complexity Analysis

The most expensive part of the algorithm is the building and evaluation of the AND/OR tree, because it is exponential in the planning horizon T, where the base of the exponent is the number of discrete levels of discretization for the demand variable D_t. However, when an AND node is added to the tree, a feasibility check is performed first: a node is added only when it meets all temporal constraints plus the demand and load constraints, and the economic dispatch associated with the node has a feasible solution.

Experimental Results

We experimented with the proposed method on a test problem adopted from (Li, Johnson, and Svoboda 1997), extended with the introduction of uncertainty in the demand.

The standard deviation of demand was assumed to be 2% of expected demand: $\sigma_t = 0.02\bar{D}_t$. No uncontrollable generators were used, so the net demand is equal to the total demand. The approximate algorithm from the previous section was implemented and compared against two existing algorithms: one of them was based on a priority list ((Wood and Wollenberg 1996)), and the other one was the decommitment algorithm proposed in (Li, Johnson, and Svoboda 1997). Our results showed that the approximate solution method provides a good balance between generation cost and risk of failure to meet demand. We performed experiments on two UC examples: one with 4 units, and another one with 20 units. We were able to calculate the truly optimal MDP solution for the 4-unit UC example, so we were able to investigate the accuracy of our approximation scheme on that problem, too. The experiments were performed on a computer with Intel Core 2 Duo E6600 CPU (2.40GHz). The algorithm was implemented in MATLAB.

Experimental Conditions

The generation cost of a committed unit i at time t is computed as a quadratic function of the produced amount of power by the unit: $f_i(x_t^i, u_t^i, d_t) = c_0^i + c_1^i p_i^t + c_2^i (p_t^i)^2$. The unit switching and start-up cost is expressed as $h(x_t^i, u_t^i, u_{t+1}^i) = tcst_i + bcst_i(1 - \exp(-\gamma x_t^i))$, if $u_t^i = 0$ and $u_{t+1}^i = 1$, and zero otherwise. In the start-up cost, the fixed component $tcst_i$ represents the cost of starting generator i, while the second term $bcst_i$ represents the cost of starting the boiler and varies exponentially with the length of the time that the unit has been off.

Under a Gaussian assumption for demand ($D_t \sim N(\bar{D}_t, \sigma_t^2)$), the risk compensation cost $g_t(u_t, d_t)$ is given by

$$\alpha' \cdot C_{FSO} \cdot \int_{\sum_i u_t^i cap_i}^{\infty} \frac{1}{\sqrt{2\pi\sigma_t^2}} exp(-\frac{(D - \bar{D}_t)^2}{2\sigma_t^2}) \cdot dD$$

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). Failure happens when the actual demand D is greater than the Maximum Committed Capacity (MCC) $\sum_i cap_i u_t^i$ of all operating units. By increasing the constant α, the weight of the risk component in the objective function is increased, thus favoring configurations with higher MCC, at the expense of a higher operational cost for running such configurations.

A 4-unit example

The decision horizon of the 4-unit UC problem was 24 hours. The coefficients $tcst_i$ and $bcst_i$ of the start-up costs for the four units were [200,2000;500,20000;100,700;44,100]. The fuel cost coefficients $[c_0, c_1, c_2]$ for the four units were [0.00211,16.51,02.7; 0.00063,21.05,1313.6; 0.00712,22.26,371.0; 0.00413,25.92,660.8], in chosen cost units. The minimum up and minimum down

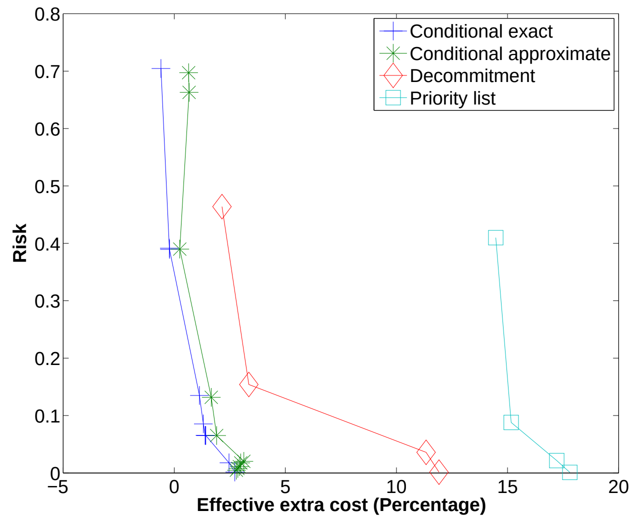

Figure 4: Performances of the algorithms on a 4-unit problem

times were $[3, 3, 2, 2]$ and $[4, 4, 3, 3]$. The minimum and maximum capacities were $[10, 10, 10, 10]$ and $[100, 90, 80, 60]$, here and henceforward, in chosen power units. The expected demand vector was $\bar{D}$ =[105,85,65,140,100,105,125,145,165,185,205,245,265,285, 200,140,100,105,125,145,165,185,205,225]. The initial operational times were $x_0 = [5, -5, 5, -5]$.

The risk versus cost curves for various methods are presented in Fig. 4. "Conditional exact" refers to the algorithm that solves the MDP exactly, i.e., all Bellman backups (Equation 6) were performed. "Conditional approximate" refers to the algorithm proposed in the previous section. In the figure, the horizontal axis is the percentage of the extra operational cost with respect to a reference operational cost, taken to be the lowest experimentally obtained operational cost for any scheduler on this problem. For this problem instance, it can be seen that the solution of the proposed algorithm is very close to optimality (the conditional exact solution), and the algorithm outperforms significantly both the priority list and the decommitment algorithms in balancing operational costs and risks. For the lowest levels of risk, which are probably close to the desired cost/risk trade-off point of an actual generation system, the loss of optimality is less than 1%, whereas the gain in costs with respect to deterministic schedulers is greater than 9%.

20-unit example

In this experiment we used all 20 generators described in (Li, Johnson, and Svoboda 1997). The expected demand vector was [2133.3, 2133.3, 2066.7, 2066.7, 2133.3, 2133.3, 2266.7, 2400.0, 2400.0, 2400.0, 2333.3, 2200.0, 2133.3, 2133.3 ,2200.0, 2266.7, 2400.0, 2400.0, 2400.0, 2400.0, 2333.3, 2200.0, 2200.0, 2066.7]. It was no longer possible to find the truly optimal conditional schedules, but it is possible to compare the performance of the conditional approx-

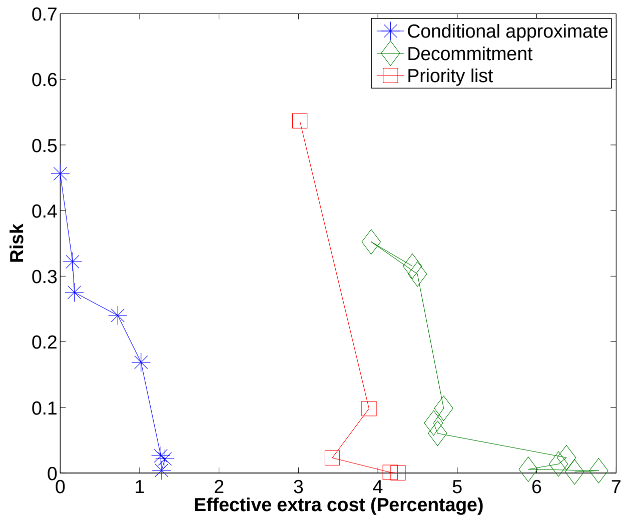

Figure 5: Performances of the algorithms on a 20-unit problem

imate, priority list, and decommitment algorithms (Fig. 5). Again, the results show that the proposed novel algorithm uniformly achieved a much better risk/cost balance than the priority list and the decommitment approaches, with operational cost savings around 4% for the lowest levels of risk.

Conclusion and Future Work

We have described a general method for representing the mixed continuous/discrete dynamics of power generation systems under multiple sources of uncertainty such as variable power demand and intermittent renewable energy sources, and have introduced a class of conditional operational plans where the unit commitment decisions are conditioned upon the state of observable random variables. The proposed factored Markov decision process models represented in the form of dynamic Bayesian networks are compact and are also easy to specify, maintain, and extend with new power sources. We have also proposed one concrete algorithm for finding such conditional operational schedules for power generation that depend on a single random variable — the net demand that aggregates in itself all sources of randomness. The algorithm focuses on small subsets of all possible configurations of generators in order to compute the schedule efficiently. Experimental results suggest that the resulting conditional plans are close to the truly optimal ones, and provide a much better trade-off between generation cost and risk of failure to meet demand than two known non-stochastic unit commitment algorithms that compute fixed schedules.

In the proposed solution algorithm, we use AND/OR trees to represent, find, and evaluate the optimal conditional plan. However, this algorithm is by no means the only possible way to solve stochastic generation problems represented by means of fMDPs and DBNs. In future work, we plan to investigate other solution methods based on approximate dynamic programming that could result in much better computational complexity. Furthermore, the current solution aggregates the variability of all stochastic variables into the net demand to the controllable power generators, for the sake of computational efficiency. This simplifies the planning problem, because the branching in the AND-OR tree is based only on that single variable. However, even higher efficiency might be possible if the conditional schedule is conditioned on the values of each individual stochastic component. This would significantly increase the complexity of the planning process, and would depend critically on finding more computationally efficient solution methods for the undrelying fMDP models.

For example, the method proposed in (Feng et al. 2004) represents the value function of the dynamic programming problem over continuous domains by adaptively discretizing such continuous variables. This approach might result in more accurate and compact representations than are possible with our method, where the tesselation of the continuous domains is performed apriori, before value functions are evaluated. Adaptive discretization is indeed compatible with our discretization scheme, too, for example by sub-dividing a simplex where the value function varies a lot (measured on its vertices), into multiple smaller simplices. The application of symbolic dynamic programming (SDP, (Sanner, Delgado, and de Barros 2011)) to the factored MDP-based formulation of the operational planning problem might be possible, too.

The formulation of the fMDP described in the paper assumes that all generators assume their intended configuration u_i^t without fail. This allows us to use the decision variables u_i^t as components of the state of the system, thus simplifying the planning process. If the possibility of equipment failure must be taken into account, the actual configuration U_i^t of the generators should be included as a random state variable in the DBN, and its probabilistic dependence on the intended configuration u_i^t can be modeled according to the failure probabilities of individual generators. Such an extension is completely compatible with the proposed modeling formalism of factored Markov decision processes.

References

Boutilier, C.; Dearden, R.; and Goldszmidt, M. 2000. Stochastic dynamic programming with factored representations. *Artificial Intelligence* 49–107.

Feng, Z.; Dearden, R.; Meuleau, N.; and Washington, R. 2004. Dynamic programming for structured continuous markov decision problems. In *Proceedings of the 20th conference on Uncertainty in Artificial Intelligence*, UAI '04, 154–161. Arlington, Virginia, United States: AUAI Press.

Li, C.; Johnson, R. B.; and Svoboda, A. J. 1997. A new unit commitment method. *IEEE Transactions on Power Systems* 113–119.

Martelli, A., and Montanari, U. 1973. Additive AND/OR graphs. In *Proceedings of the Third International Joint Conference on Artificial intelligence*, 1–11.

Nikovski, D., and Esenther, A. 2011. Construction of embedded Markov decision processes for optimal control of non-linear systems with continuous state spaces. In *IEEE Conference on Decision and Control and European Control Conference*, 7944–7949.

Nikovski, D.; Lidicky, B.; Zhang, W.; Kataoka, K.; and Yoshimoto, K. 2012. Markov decision processes for train run curve optimization. In *Electrical Systems for Aircraft, Railway and Ship Propulsion (ESARS), 2012*, 1–6.

Nikovski, D.; Xu, J.; and Nonaka, M. 2013. A method for computing optimal set-point schedules for HVAC systems. In *Proceedings of the 11th REHVA World Congress CLIMA 2013*.

Preparata, F. P., and Shamos, M. I. 1990. *Computational Geometry*. Heidelberg: Springer Verlag.

Puterman, M. L. 1994. *Markov Decision Processes—Discrete Stochastic Dynamic Programming*. New York, NY: John Wiley & Sons, Inc.

Sanner, S.; Delgado, K. V.; and de Barros, L. N. 2011. Symbolic dynamic programming for discrete and continuous state MDPs. In *Proceedings of the 27th conference on Uncertainty in Artificial Intelligence*, UAI '11, 643–652. Arlington, Virginia, United States: AUAI Press.

Takriti, S.; Birge, J. R.; and Long, E. 1996. A stochastic model of the unit commitment problem. *IEEE Transactions on Power Systems* 1497–1508.

Wood, A. J., and Wollenberg, B. F. 1996. *Power Generation, Operation, and Control*. New York: Wiley-Interscience.

Xia, X., and Elaiw, A. M. 2010. Optimal dynamic economic dispatch of generation: a review. *Electric Power Systems Research* 975–986.

Xia, X.; Zhang, J.; and Elaiw, A. 2011. An application of model predictive control to the dynamic economic dispatch of power generation. *Control Engineering Practice* 19(6):638–648.