Chance-Constrained Outage Scheduling using a Machine Learning Proxy

Gal Dalal*, Elad Gilboa*, Shie Mannor*, and Louis Wehenkel†
*Department of Electrical Engineering, Technion, Israel, {gald,egilboa}@tx.technion.ac.il, shie@ee.technion.ac.il
†Montefiore Institute – Department of Electrical Engineering and Computer Science
University of Liège, L.Wehenkel@ulg.ac.be

***Abstract*—Outage scheduling aims at defining, over a horizon of several months to years, when different components needing maintenance should be taken out of operation. Its objective is to minimize operation-cost expectation while satisfying reliability-related constraints. We propose a data-driven distributed chance-constrained optimization formulation for this problem. To tackle tractability issues arising in large networks, we use machine learning to build a proxy for predicting outcomes of power system operation processes in this context. On the IEEE-RTS79 and IEEE-RTS96 networks, our solution obtains cheaper and more reliable plans than other candidates. All our code (matlab) is publicly available at https://github.com/galdl/outage_scheduling.**

***Index Terms*—Outage Scheduling, Stochastic Optimization, Scenario Optimization, Chance Constraints**

I. Nomenclature

$\mathcal{L}$	Set of components requiring an outage
$\mathcal{T}_m$	Candidate outage moment set
$u_m \in \mathcal{U}_{RT}$	Outage schedule
T	Outage scheduling planning horizon
C	Outage scheduling cost function
$r/\overline{r}$	Momentary/average reliability
$r_{\min}$	Minimal tolerable reliability
$\mathrm{LS}/\overline{\mathrm{LS}}$	Momentary/average load shedding
$\mathrm{LS}_{\max}$	Maximal tolerable load shedding
$\alpha_{\mathrm{r}}/\alpha_{\mathrm{LS}}$	Reliability/load-shedding criterion probability tail
$u_s \in \mathcal{U}_s$	Short-term (unit commitment) decision
C_s	Short-term cost
$u_{\mathrm{RT}} \in \mathcal{U}_{RT}$	Real-time action
C_{RT}	Real-time cost
$S_t \in \mathcal{S}$	System state at time t
$Z = \{S_t\}_{t=1}^{T}$	Stochastic scenario
y_s/y_{RT}	Short-term/real-time informational state
$h(u_m)$	Outage schedule feasibility constraints
$n^{(t)}/n^{(b)}$	Number of transmission-lines/buses
$n^{(g,d)}/n^{(g,w)}$	Number of dispatchable/wind generators
$W_{\mathrm{da}}/D_{\mathrm{da}}$	Day-ahead wind-generation/load forecast
W_t/D_t	Wind-generation/load at time t
J_t	Seasonal weather factor at time t
top_t	Topology at time t
$t_m/t_s/t_{\mathrm{RT}}$	Mid/short/real-time time index
T_s/T_{RT}	Short-term/real-time simulation window length
N_s/N_{RT}	Short-term/real-time simulation replicates

II. Introduction

Outage scheduling is performed by transmission system operators (TSOs), as an integral part of asset management, in order to carry out component maintenance and replacement activities [1]. However, scheduling of the required outages for maintenance jobs is a complex task since it must take into account constrained resources (e.g. working crews, hours, and budget), increased vulnerability of the grid to contingencies during outages, and the impact of the scheduled outages on operations. Moreover, outage schedules, which are planned on a mid-term scale of several months to years, must also be robust with respect to uncertainties.

In this work, we present a general framework for assessing the impact of a given outage schedule on expected costs and system reliability, incurred while operating the system during the schedule's period. In addition, we formulate and solve a stochastic optimization program for optimally scheduling a list of required outages. To do so, we take into account the smaller-horizon decision processes taking place during this time interval. These latter concern day-ahead operational planning and real-time operation.

The complex dependence between the multiple time-horizons and the high uncertainty in the context of mid-term planning renders the corresponding assessment problem challenging. As demonstrated in [2], solving an extensive amount of unit commitment (UC) problems to mimic short-term decision-making does not scale well to realistic grids, with thousands of buses or more. This is specifically burdensome while simulating trajectories of months to years. To deal with this complexity we propose to use machine learning to design a proxy that approximates short-term decision making, relieving the dependence of mid-term outcome assessment on accurate short-term simulations; ergo, allowing a tractable assessment method. Specifically, we replace exact UC computations with pre-computed UC solutions to problems with similar input conditions. See Section VI for further details on the method.

When planning for future outages to enable maintenance, a certain reliability criterion is attempted to be satisfied at all future times. Nowadays, TSOs often ensure the deterministic N-1 reliability criterion per each post-outage scenario, while other probabilistic criteria are also being investigated [3], [4]. To make the system secured months in advance, the asset management operator should ideally assess whether each of the possible future scenarios is secure, by taking into account

the coordination with day(s)-ahead and (intra)hourly operation. Since considering all possible realizations of future events is impractical, they must be approximated using sampled paths of future scenarios. In this work, we thus also devise a sampling scheme that richly represents possible occurrences while being tractable. We trust such methods are crucial for high-quality mid-term analysis.

A. Related Work

Current practice in transmission system asset management offers three main approaches: time-based, condition-based, and reliability-centered preventive maintenance [5]. As for the academic literature, two popular trade-offs are i) increasing equipment reliability via maintenance while minimizing maintenance costs, and ii) minimizing the effect of outages on socio-economic welfare while satisfying operating constraints.

In [6], the first above trade-off was considered in a two-stage formulation. The first stage schedules mid-term maintenance that imposes conditions on the second stage problem: short-term N-1 secured scheduling. By choosing a maintenance action per each time-interval, Weibull asset failure probability was analytically minimized. In [7], an analytic objective function was also designed. There, maintenance reduced cumulative risk of events such as overloads and voltage collapses, assuming known year-ahead generation and load profiles. The accumulated gain was negative during the actual maintenance, but positive afterwards due to its failure rate reduction. In more recent work [8], a greedy outage scheduling algorithm used Monte-Carlo simulations to assess the impact of outages on system operation. By mimicking experts heuristics for mid-term outage-scheduling, it enables long-term assessment of system development and maintenance policies.

As mentioned earlier, coordination with UC and economic dispatch may render the outage schedule assessment intractable, especially under security criteria. To overcome this, a coordination strategy between the different tasks was proposed in [9]. There, mid-term planning over a deterministic 168-hour-long scenario minimized UC scheduling costs under changing network topology. In [10], a stochastic coordination model is formulated as an hourly Mixed Integer Program (MIP) using scenario trees that involve various disturbances. Lagrange Relaxation is then applied to decompose and separately solve the long-term maintenance and short-term deterministic UC problems. A resembling formulation and approach appear in [11]. In the bilevel outage scheduling model there, a yearly upper-level transmission-capacity-margin objective is constrained by lower-level market clearing problems. Using equilibrium constraints, the problem is recast as a mixed integer-linear program (MILP) and solved with branch-and-cut techniques.

B. Our Contribution

Our contribution is three-fold. First, we provide a new probabilistic mathematical framework that accounts for three entities involved in the multiple-horizon decision-making process; these are namely the mid-term, short-term and real-time decision makers. Our model captures their hierarchy and formulates their coordination using an information sharing scheme, that limits each via partial observability.

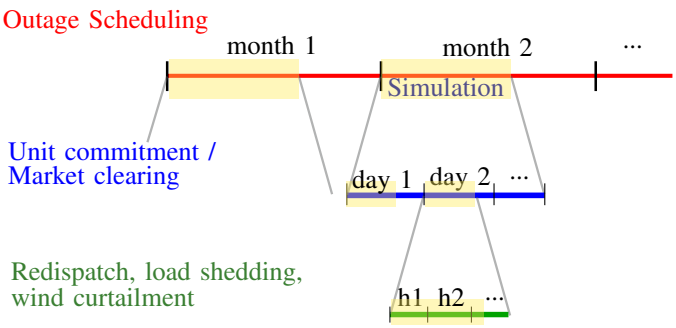


Fig. 1: Our *Bayesian hierarchical window sampling* is a scenario generation approach, which combines both sequential trajectory simulation and snapshot sampling. In each level of the hierarchy, a snapshot of future grid and environment conditions is sampled, and sequential simulation is performed from that point on, for a limited time window.

Second, we introduce a component that greatly reduces simulation runtime by predicting approximated short-term decision outcomes; we refer to it as a *proxy*. We do so with a well-known machine learning algorithm, *nearest neighbor*. Machine learning has been applied previously to various power system applications, such as power flow prediction [12], [13], [14], [15], disturbance detection [16], and fault classification [17]. To the best of our knowledge, we are the first to apply it for predicting UC outcomes. But more significantly, this work is novel in integrating a proxy to replace a decision making layer in long-term planning hierarchy, who takes inputs from an upper layer and its outputs are used in a lower layer. This enables a critical reduction in computation time, which turns the table and deems large-scale data-driven assessment with multiple time-horizons tractable.

Third, we devise a scenario-based optimization methodology. It builds on our scenario generation approach, *Bayesian hierarchical window sampling*, which combines both sequential trajectory simulation and snapshot sampling while accounting for coordination between the three decision layers; see Fig. 1. Using it, we solve our stochastic chance-constrained outage scheduling formulation for IEEE-RTS79 and IEEE-RTS96 with distributed computing and show promising results.

The individual merits of the above methods are given in detail in the rest of the paper but are also summarized as follows. The hierarchical window sampling approach is decomposed for supporting multiple stakeholder situations. It also allows for natural top-down hierarchical simulation given conditional empirical distributions of parameters such as daily wind generation and hourly deviations, which are usually easier to represent compared to joint distributions. Next, our machine-learning UC algorithm allows for more tractable simulation by re-using pre-computed UC solutions instead of accurately computing them each time from scratch. Lastly, the distributed Cross-Entropy optimization method enables searching the combinatorial space of outage schedules based on Monte-Carlo simulation, without requiring derivable and

decomposable analytic forms of objective and constraints.

C. Comparison of Our Methodology to Existing Literature

Traditionally, for both short and long time horizons, the power system literature often formulates MIPs and tractably solves them using relaxation and decomposition methods such as the respective Lagrange Relaxation [18] and Benders Decomposition [19]. This is also the case with literature considered state-of-the-art for outage scheduling [10], [11].

A potentially intriguing direction would have been a direct comparison via simulation of the above work to the one presented here. However, this is impractical because both [10], [11] and other related works require the explicit probabilities of each event and a closed mathematical (well-structured) form of the corresponding cost and reliability criteria. With our data-driven methodology neither of the above is accessible, because of the way real-time simulation is conducted. Indeed, in our case, probabilities of the short-term inputs to the UC can be explicitly computed in principle; however, the consequent real-time scenarios and their probabilities are dependent on the outcome of these UC programs. Specifically, the real-time OPF depends on the day-ahead plan in terms of generator availability, planned load shedding and planned wind curtailment. Put differently, the involved probabilities are transformed from the UC input space to its output, so they cannot be expressed analytically. Similarly, and perhaps even more problematically, the real-time reliability criterion defined here depends on the solutions of numerous non-convex ACPFs (see Section III-B). So closed form expressions for this criterion are not accessible. Therefore, we compare the traditional line of work as in [10], [11] to ours with by summarizing the differences as follows.

Traditional analytic MIP formulations are common both in literature and industry. It benefits from well-established MIP optimization theory. Also, recent uncertainty-aware formulations for short-term operation could be potentially leveraged for long-term as well. On the other hand, this methodology suffers from the several shortcomings. First, strong assumptions on probabilities and implications are made; the analytic MIP approach assumes some fixed scenario set with known event probability along with its price and reliability implications. Contrarily, our method is data-driven. It solely relies on access to some black-box scenario generator; this even conforms with resampling existing real-life data (in our work we sample a distribution that is known to us, but this is not an actual restriction). Generating more scenarios during the optimization itself, if needed to improve accuracy online, is done on-the-go by sampling. Second, efficient MIP relaxation and decomposition methods necessitate suitable mathematical structures. Contrarily, our approach is fully Monte-Carlo based and hence poses no limitations in terms convexity or other mathematical structures. Any computable criterion of interest can be assessed and optimized. Adding layers such as real-time operation is also convenient due to the modular structure. Lastly, the modular structure allows replacing layers in the hierarchy with approximated surrogates, i.e., proxies such as the one introduced in this work. It is not clear whether this can be done as part of a MIP formulation.

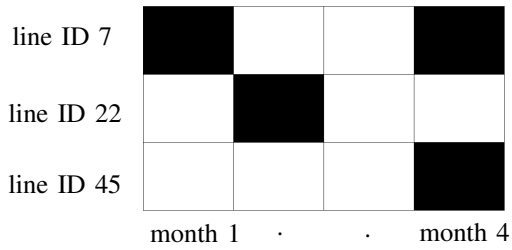


Fig. 2: Toy example of an outage schedule u_m. The black and white entries correspond to 1 and 0. In this example, the planning horizon is 4 months and transmission lines with IDs $\{7, 22, 45\}$ are required to undergo outages; two outages are required for line 7, and 1 for lines $22, 45$. In this specific schedule, line 7 outages are scheduled for months 1 and 4, line 22 outage is scheduled for month 2, etc.

III. Mathematical Problem Formulation

In our problem setting, the TSO lists necessary outages, each one defined by a duration and a specific component to be taken out of operation for maintenance. A specific component can be required to undergo more than a single outage. The outage scheduling horizon (e.g. several months, or a couple of years) is split into T hourly time-steps. Let $\mathcal{L}$ be the set of components required to undergo a single outage or more, and $\mathcal{T}_m$ be a given set of candidate outage moments (e.g. monthly steps). The decision variable $u_m \in \mathcal{U}_m = \{0,1\}^{|\mathcal{L}\times\mathcal{T}_m|}$ is a matrix, with $[u_m]_{\ell,i} \in \{0,1\}$ denoting whether component $\ell \in \mathcal{L}$ undergoes an outage that starts at time $i \in \mathcal{T}_m$. An example for an outage schedule is given in Fig. 2.

We formulate the mid-term stochastic optimization program in (1), where the goal is to minimize expected future operating costs by adopting an optimal planned outage schedule u_m.

$$\min_{u_m\in\mathcal{U}_m} \sum_{t=1}^{T} \mathbb{E}_{S_t\in Z}\left\{C(S_t, u_m, u_s^*, u_{\text{RT}}^*)\right\} \tag{1a}$$

$$\text{s.t.}\quad \mathbb{P}\left\{\overline{r}(Z, u_m, u_s^*, u_{\text{RT}}^*) \geq r_{\min}\right\} \geq 1-\alpha_{\text{r}} \tag{1b}$$

$$\mathbb{P}\left\{\overline{\text{LS}}(Z, u_m, u_s^*, u_{\text{RT}}^*) \leq \text{LS}_{\max}\right\} \geq 1-\alpha_{\text{LS}} \tag{1c}$$

$$h(u_m) \leq 0 \tag{1d}$$

$$u_s^* = \underset{u_s\in\mathcal{U}_s(u_m)}{\arg\min}\ C_s(u_s, y_s, u_m) \tag{1e}$$

$$u_{\text{RT}}^* = \underset{u_{\text{RT}}\in\mathcal{U}_{\text{RT}}(u_s^*, u_m)}{\arg\min}\ C_{\text{RT}}(u_{\text{RT}}, y_{\text{RT}}, u_s^*, u_m). \tag{1f}$$

This formulation's components are explained as follows.

A. Objective

The objective in (1a) is the aggregated expected cost of operational decisions summed over the evaluation horizon. The expectation is w.r.t. the distribution of the uncertain future conditions of the grid encased in and denoted by stochastic exogenous scenario Z. Scenario $Z = (S_1, \ldots, S_T)$ is a series of states S_t, which are introduced in more detail in Section IV-A. When making decisions in the mid-term time-horizon, one must take into account the smaller-horizon decisions that take place during it. In this work,

the smaller-horizon decisions considered are short-term (day-ahead) operational planning $u_s \in \mathcal{U}_s(u_m)$ and real-time control $u_{\text{RT}} \in \mathcal{U}_{\text{RT}}(u_s)$. Each of the sets of candidate smaller-horizon decisions $\mathcal{U}_s(u_m),\ \mathcal{U}_{\text{RT}}(u_s, u_m)$ is function of the decisions that are taken higher in the hierarchy.

In our work, a real-time decision u^*_{RT} defines a vector of redispatch values for each redispatchable generator, wind curtailment values for each wind generator, and LS values for each bus, and is determined by minimizing the cost C_{RT} of deviation from the day-ahead market schedule u^*_s. The latter is determined by minimizing a day-ahead objective C_s. Lastly, we choose the function $C(S_t, u_m, u^*_s, u^*_{\text{RT}})$ to only account for the real-time operating costs. Specifically, it is identical to C_{RT}, apart for the LS cost. This is because LS is already addressed via (1c). However, the formulation is general, and may in principle also incorporate real-time LS, as well as market surplus and day-ahead reserve purchase costs, if deemed important.

B. Reliability and Load Shedding Chance-Constraints

To maintain the generality and flexibility of our model while respecting its probabilistic framework, we define a reliability metric that is independent of smaller time-horizon specificities. It allows for equitable comparison between different maintenance strategies and operation policies. Inspired by the common N-1 criterion used in the industry, we adapt it to our probabilistic setup. Namely, we consider the system's ability to withstand any contingency of a single component. We thus define reliability as the portion of contingencies under which the system retains safe operation, which, practically, we measure via AC power flow convergence.

For this, denote by $\mathcal{N}_{-1}(S_t, u_m)$ the N-1 contingency list and by $r(S_t, u_m, u^*_s, u^*_{\text{RT}}) \in [0, 1]$ the real-time reliability, which for brevity we also denote by $r(S_t, u_m)$. The latter is calculated for given state S_t and is dependent of current topology, dictated by u_m:

$$r(S_t, u_m) = \frac{1}{|\mathcal{N}_{-1}(S_t, u_m)|} \sum_{c \in \mathcal{N}_{-1}(S_t, u_m)} \mathbb{I}_{[\text{PF}(c, S_t, u_m)]},$$

where $\mathbb{I}_{[\text{PF}(c, S_t, u_m)]}$ equals 1 if a feasible ACPF solution exists during contingency $c \in \mathcal{N}_{-1}$, and 0 otherwise. Also, let

$$\overline{r}(Z, u_m, u^*_s, u^*_{\text{RT}}) = \frac{1}{T} \sum_{S_t \in Z} r(S_t, u_m); \tag{2}$$

i.e., the average success rate for scenario Z.

In similar fashion, let $\text{LS}(S_t, u_m)$ be the total load shed in state S_t, as determined by the real-time decision u^*_{RT}, and $\overline{\text{LS}}(Z, u_m, u^*_s, u^*_{\text{RT}}) = \frac{1}{T} \sum_{S_t \in Z} \text{LS}(S_t, u_m)$; i.e., the average amount of load shed during scenario Z. This LS criterion in fact corresponds to a known index: expected demand not supplied (EDNS), up to normalization by the horizon length.

Based on the these definitions, the chance-constraints in (1b)-(1c) ensure that the average reliability remains above a minimal value $r_{\min}$ and that the average LS remains below a maximal value $\text{LS}_{\max}$, with respective probabilities $1 - \alpha_{\text{r}}$ and $1 - \alpha_{\text{LS}}$. Based on reasonable achievable values for our specific test-case modifications listed in Section VIII, throughout this work we set $r_{\min} = 0.8, \text{LS}_{\max} = 0.5\%$ of overall load capacity, $\alpha_{\text{r}} = 0.05,\ \alpha_{\text{LS}} = 0.05$.

The reason for explicitly incorporating the two chance constraints together is to ensure both high reliability and low LS at the same time, as these two obviously trade-off. We further relate to this trade-off in Section VIII.

C. Feasibility Constraints

Maintenance feasibility constraints $h(u_m)$ in (1d) define which maintenance schedules are feasible, e.g., cannot maintain more than two assets per month.

D. Coordination with Smaller-Horizon Subproblems

Lastly, the constraints in (1e)-(1f) ensure coordination between mid-term and smaller-horizon decisions. The informational states y_s and y_{RT} appearing as arguments of $C,\ C_s,\ C_{\text{RT}}$ depict the partial information revealed to the respective decision makers in these time-horizons; further details on the notion of informational states are given in Section IV-B.

IV. Decision Making Model

In this section, we elaborate on our probabilistic decision making model. We define a state-space representation encapsulating all exogenous uncertain information and the decision makers' limited access to this information. Our model is generic and can be adapted for additional uncertain factors.

A. State-Space

State $S_t \in \mathcal{S}$ captures all exogenous uncertain information at time t, required to make informed decisions in all considered time-horizons. Let $n^{(t)},\ n^{(b)},\ n^{(g,d)},\ n^{(g,w)}$ respectively be the number of transmission lines, buses, dispatchable generators, and wind generators in the network. The state S_t is defined as the following tuple:

$$S_t = (J_t, W_{\text{da}}, D_{\text{da}}, W_t, D_t, \text{top}_t), \text{where}$$

- $J_t \in \mathbb{R}^2$ is the seasonal weather factor, determining the intensity of demand and wind generation. This variable changes monthly, with values drawn around a mean profile corresponding to typical seasonal trends.
- $W_{\text{da}} \in \mathbb{R}_+^{n^{(g,w)} \times T_{\text{da}}}$ is the day-ahead wind generation forecast, where T_{da} is the day-ahead planning horizon (24 in our simulations). Notice that variables with subscript 'da' remain fixed for time periods of length T_{da}, and are updated each T_{da} time-steps.
- $D_{\text{da}} \in \mathbb{R}_+^{n^{(b)} \times T_{\text{da}}}$ is the day-ahead load forecast.
- $W_t \in \mathbb{R}_+^{n^{(g,w)}}$ is the realized wind generation at time-step t. It is assumed fixed during the intra-day interval (1 hour).
- $D_t \in \mathbb{R}_+^{n^{(b)}}$ is the realized load at time-step t.
- $\text{top}_t \in \{0, 1\}^{n^{(t)}}$ is the network transmission line topology at time-step t, as determined by exogenous events. Entry $\text{top}_t(i) = 0$ indicates line i is offline at time t, due to a random forced outage.

B. Informational States

Decision makers with different time-horizons are exposed to different degrees of information; i.e., the higher the decision's temporal resolution, the more state variables are realized at the time of the decision. We formulate these granularities via *informational states* as follows. Denote $S_t^{1:k}$ to be S_t's sub-vector containing entries 1 to k. Let $y_s = S_t^{1:3} = (J_t, W_{\text{da}}, D_{\text{da}})$ and $y_{\text{RT}} = S_t$; these are respectively the short-term and real-time informational states. When performing her decision, the short-term planner is exposed to y_s, which carries the realizations of the day-ahead generation and load forecasts. Notice he is also exposed to the higher-level mid-term decision u_m; however, we do not model it as a part of the informational state as it is not exogenous. As for the real-time operator, he is exposed to realized values of all state entries, i.e. y_{RT}, and is similarly informed of the higher level decisions u_m and u_s.

For completeness, we also define the mid-term informational state $y_m = S_t^1 = J_t$, even though it does not appear directly in (1) (it does appear later for scenario generation purposes). Notice that in our work u_m is an open-loop mid-term strategy, so y_m is not used to revise mid-term decisions.

C. Smaller-horizon Formulations

Our formulation contains three hierarchical levels of decision making, namely mid-term outage scheduling, short-term day(s)-ahead planning, and (intra)hourly real-time control. We often refer to the short-term and real-time problems as the *inner subproblems*. We now present the candidate decisions in these latter.

1) Short-term Formulation: The formulation

$$u_s^* = \underset{u_s \in \mathcal{U}_s(u_m)}{\arg\min} \quad C_s(u_s, y_s, u_m), \tag{3}$$

which also appears in (1e), is set in this work to be UC. As explained in Section III, we choose the cost C in (1a) to be solely based on real-time realizations and decisions. Thus, the UC cost here is not to be minimized by the mid-term planner; rather, the UC solution is plugged into the real-time problem for setting commitment constraints and redispatch costs reference. Notice the UC formulation depends on day-ahead forecasts of wind power and load $W_{\text{da}}, D_{\text{da}}$. These are parts of the informational state y_s, to which the decision maker is exposed when facing his day-ahead planning decision. The feasible action-space $\mathcal{U}_s(u_m)$ in (3) depends on the topology set by the mid-term decision u_m, and it may also embody a reliability criterion of choice, e.g. N-0 or N-1.

We now bring our complete UC formulation. In this work, the reliability criterion used is N-0; i.e., no contingency list is considered at the subproblem level. Instead, our probabilistic notion analogous to N-1 resiliency is ensured via (1b). Also, we use the DCPF approximation, in which voltage magnitudes and reactive powers are eliminated from the problem, and real power flows are modeled as linear functions of the voltage angles [20]. This results in the following MILP which we model with YALMIP [21] and solve with CPLEX [22]. The formulation and notations rely on [20], with two main differences: i) we extended it to the time domain, and ii) we added wind-curltaiment and load-shedding penalties.

$$u_s^* = \underset{u_s \in \mathcal{U}_s(u_m)}{\arg\min}\ C_s(u_s, y_s, u_m) = \underset{\alpha, \Theta, P_{g,t}, \text{WC}, \text{LS}}{\arg\min}$$
$$\sum_{t=1}^{T_{\text{da}}} \left[\sum_{i=1}^{n^{(g,d)}} \left(\alpha_t^{(i)} f_P^{(i)}(P_{g,t}^{(i)}) + \alpha_t^{(i)}(1-\alpha_{t-1}^{(i)}) \text{SU}_i(t_{\text{off}}^{(i)}(\alpha,t)) \right) \right.$$
$$\left. + \sum_{i_w=1}^{n^{(g,w)}} \text{WC}_t^{(i_w)} \cdot C_{\text{WC}} + \sum_{i_b=1}^{n^{(b)}} \text{LS}_t^{(i_b)} \cdot \text{VOLL} \right], \tag{4a}$$

subject to (4b)

$$g_{P,t}(\Theta, \alpha, P_g) = B_{\text{bus}}\Theta_t + P_{\text{bus,shift}} + D_{\text{da}} + G_{\text{sh}} - \text{LS}_t - (W_{\text{da}} - \text{WC}_t) - C_g(\alpha_t . * P_{g,t}) = 0, \tag{4c}$$

$$B_f \Theta_t + P_{f,\text{shift}} - F_{max} \leq 0, \tag{4d}$$

$$-B_f \Theta_t - P_{f,\text{shift}} - F_{max} \leq 0, \tag{4e}$$

$$\theta_{i,t} = \theta_i^{\text{ref}} \quad i \in \mathcal{I}_{\text{ref}}, \tag{4f}$$

$$\alpha_t^{(i)} P_g^{(i),\min} \leq P_{g,t}^{(i)} \leq \alpha_t^{(i)} P_g^{(i),\max}, \quad i = 1, \ldots, n^{(g,d)}, \tag{4g}$$

$$0 \leq \text{WC}_t^{(i_w)} \leq W_{\text{da},t}^{(i_w)}, \quad i_w = 1, \ldots, n^{(g,w)}, \tag{4h}$$

$$0 \leq \text{LS}_t^{(i_b)} \leq D_{\text{da},t}^{(i_b)}, \quad i_b = 1, \ldots, n^{(b)}, \tag{4i}$$

$$t_{\text{off}}^{(i)}(\alpha, t) \geq t_{\text{down}}^{(i)}, \quad i = 1, \ldots, n^{(g,d)}, \tag{4j}$$

$$t_{\text{on}}^{(i)}(\alpha, t) \geq t_{\text{up}}^{(i)}, \quad i = 1, \ldots, n^{(g,d)}, \tag{4k}$$

$$t = 1, \ldots, T_{\text{da}}. \tag{4l}$$

The formulation's components are explained as follows.

- $\alpha \in \{0,1\}^{n^{(g,d)} \times T_{\text{da}}}$ denotes commitment (on/off) status of all dispatchable generators at all time-steps.
- $\Theta \in [-\pi, \pi]^{n^{(b)} \times (n^{(t)}+1) \times T_{\text{da}}}$ denotes voltage angle vectors for the N-1 network layouts at all time steps.
- $P_g \in \mathbb{R}_+^{n^{(g,d)} \times T_{\text{da}}}$ denotes the dispatchable generation vector, with f_P being its piecewise-linear cost function.
- $\text{WC} \in \mathbb{R}_+^{n^{(g,w)} \times T_{\text{da}}}, \text{LS} \in \mathbb{R}_+^{n^{(b)} \times T_{\text{da}}}$ denote the wind curtailment and LS decision vectors, with $C_{\text{WC}}, \text{VOLL}$ being their corresponding fixed prices per MW.
- $t_{\text{down}}^{(i)}, t_{\text{up}}^{(i)}$ denote minimal up and downtime limits for generator i, after it had been off/on for $t_{\text{off}}^{(i)}/t_{\text{on}}^{(i)}$; the latter are functions of α and t, as depicted in (4a).
- $\text{SU}_i(t_{\text{off}}^{(i)}(\alpha, t))$ denotes start-up cost of dispatchable generator i after it had been off for $t_{\text{off}}^{(i)}$ time-steps.
- $g_{P,t}(\Theta, \alpha, P_g)$ denotes the overall power balance equation for line l being offline.
- $B_{\text{bus}}, P_{\text{bus,shift}}$ denote nodal real power injection linear coefficients.
- $B_f, P_{f,\text{shift}}$ denote line flow linear coefficients.
- G_{sh} denotes a vector of real power consumed by shunt elements.
- C_g denotes generator-to-bus connection matrix, where $(\alpha_t . * P_g)$ denotes the dot-product of the two vectors.
- F_{max} denotes line flow limits.
- $\mathcal{I}_{\text{ref}}$ denotes the set of indices of reference buses, with θ_i^{ref} being the reference voltage angle.
- $P_g^{(i),\min}, P_g^{(i),\max}$ denote minimal and maximal power outputs of generator i.

Furthermore,

- (4c)-(4e) ensure load balance and network topology constraints;
- (4f)-(4i) restrict the decision variables to stay within boundaries. Namely, voltage angle limits, generator minimal and maximal power output range, wind curtailment and LS limits; and
- (4j)-(4k) bind the different time steps to follow generator minimal up and downtime thermal limits.

2) Real-time Formulation: In real-time, deviations from the forecasts and other unforeseen events need to be rectified. The resulting adaptation decisions are taken on an (intra-)hourly basis. The formulation

$$u^*_{\text{RT}} = \underset{u_{\text{RT}} \in \mathcal{U}_{\text{RT}}(u^*_s, u_m)}{\arg\min} \quad C_{\text{RT}}(u_{\text{RT}}, y_{\text{RT}}, u^*_s, u_m), \tag{5}$$

which also appears in (1f), follows the UC solution in (3) as a baseline. In our work, we model and account for generation adjustments, as well as LS and wind curtailment. Additional adjustments that go beyond the scope of this work can be, e.g., grid topology alterations, and flexible load reduction. Problem (5) is solved sequentially for each hour, where each solution at time t is fed to the next one at time $t+1$ to incorporate temporal constraints. Similarly as in (3), a reliability criterion of choice, such as N-1, may be ensured via the definition of the set $\mathcal{U}_{\text{RT}}(u^*_s, u_m)$. Although here, as in the short-term formulation, we do not employ such a criterion due to the probabilistic approach we take in Problem (1).

Our real-time formulation is identical to the short-term planning formulation (4) except for the following differences:

- The real-time horizon is 1, i.e., $T_{\text{da}} = 1$ instead of 24 in the day-ahead horizon. It is thus solved independently per each hour of day and passes required information between consecutive hours.
- Wind power and load forecasts $W_{\text{da}}, D_{\text{da}}$, which are matrices with T_{da} columns, are replaced with their respective single-column hourly realizations W_t, D_t.
- The on/off commitment schedule α^* is no longer a decision variable, rather it is retrieved from the short-term solution and plugged-in as a constant.
- If a solution cannot be found to the single-hour convex program, rescheduling takes place; i.e., the single-hour program is resolved, this time with α_t being a decision variable, as in original the day-ahead formulation (4).
- A symmetric redispatch cost is added to the objective: $\sum_{i=1}^{n^{(g,d)}} \alpha_t^{*,i} |f_P^{(i)}(P_{g,t}^{*,(i)}) - f_P^{(i)}(P_{g,t}^{(i)})|$. It juxtaposes the day-ahead hourly generation plan $P_{g,t}^{*,(i)}$ with the real-time hourly generation value $P_{g,t}^{(i)}$.

V. Scenario Generation

To solve (1) in the face of exogenous uncertainties and the intricate interaction between these uncertainties, we rely in this work on scenario-based simulation [23]. Existing literature on scenario generation splits into two main categories. The first is full-trajectory simulation [7], where (intra)hourly developments are simulated as a single long sequence. In our mid-term problem that spans over a whole year, such an approach will result in high variance and possibly necessitate an intractable number of samples to produce a decent evaluation of scenario costs. The second category of approaches is based on snapshot sampling of static future moments [24]. The main issue with this methodology is the loss of temporal information.

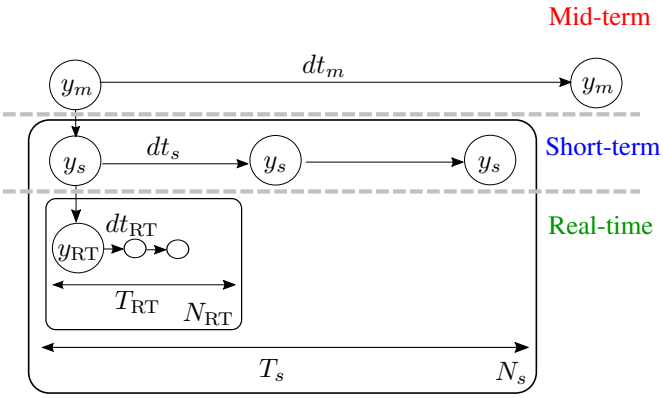


Fig. 3: Our Bayesian hierarchical window scenario sampling approach for scenario generation relies on a conditional factorization of the state to its three informational states.

In light of this, we introduce a new scenario generation approach, *Bayesian hierarchical window scenario sampling*, which is a hybrid of the two aforementioned methodologies, aimed at mitigating the disadvantages of each of them. Relying on Bayesian factorization, we decompose of the probability of state S_t to

$$\mathbb{P}\{S\} = \mathbb{P}\{y_{\text{RT}}|y_s, y_m\}\,\mathbb{P}\{y_s|y_m\}\,\mathbb{P}\{y_m\},$$

where the time index was stripped away for brevity.

Notice that since each of the real-time and short-term processes are conditioned on higher levels in the hierarchy, the state sequence S_t is a stationary Markov process; i.e.,

$$\mathbb{P}\{Z\} = \mathbb{P}\{S_0\} \cdot \mathbb{P}\{S_1|S_0\} \ldots \mathbb{P}\{S_T|S_{T-1}\},$$

where $\mathbb{P}\{S_{t+1}|S_t\}$ is a stationary state transition probability.

With that in mind, we visualize our sampling process in Fig. 3 and describe it as follows. First, we draw monthly parameters for wind and load intensity, i.e., draw a sequence $\{y_m^{(t_m)}\}$ from transition probability $\mathbb{P}\left\{y_m^{(t_{m+1})}|y_m^{(t_m)}\right\}$, where t_m is a monthly time index. Then, we draw N_s replicas of T_s consecutive days[1]; this results in sequences $\{y_s^{(t_s)}\}$ drawn from transition probability $\mathbb{P}\left\{y_s^{(t_{s+1})}|y_s^{(t_s)}, y_m^{(t_m)}\right\}$, where t_s is a daily time index. Lastly, per each such day, we draw N_{RT} replicas of T_{RT} consecutive hours and form sequences $\{y_{\text{RT}}^{(t)}\}$ from transition probability $\mathbb{P}\left\{y_{\text{RT}}^{(t+1)}|y_{\text{RT}}^{(t)}, y_s^{(t_s)}, y_m^{(t_m)}\right\}$. Having realizations of day-ahead forecasts in y_s and their corresponding hourly realizations in y_{RT}, we can respectively solve the daily and hourly inner subproblems. Based on

[1]The choice of the window lengths T_s and T_{RT} controls the level of arbitration between sequential scenario sampling and static snapshot sampling. Essentially, they arbitrate between the bias and variance of the sampling process. Completely sequential trajectory sampling has low bias but high variance, while completely static snapshot sampling lowers the variance, though it introduces bias due to its simplicity and choice of times of snapshots.

the incurred costs, we are able to evaluate the scenarios' accumulated costs per each month in a parallel fashion.

A. Comparison to Quasi-Static Time Series

We now discuss the relation between our scenario generation approach and an alternative – quasi-static time series (QSTS). Similarly to our simulation, QSTS also performs sequential steady-state power flow calculations. In its naive implementation [25], this method corresponds to the fully sequential simulation discussed earlier in this section. Recently, more sophisticated variants of it have been developed. One relevant example is [26], in which both long and short time-steps are simulated: coarse simulation is performed by default and when a material state change is detected, finer simulation takes place, starting again from the last long timestep index. Both our technique and the one in [26] focus on improving computational tractability of long-horizon simulation. However, the latter is designed for flat simulation, as opposed to the hierarchical multi-horizon setup we tackle in our work. Due to the complexity involved by considering three stakeholders and their state evolution, we decompose the state itself and not only its temporal resolution. Also, since our framework is probabilistic, we rely on Bayesian conditional factorization. This allows sampling multiple possible sequential evolutions given a specific realization of month/day. Inspiring from [26], one can analogously make the number of samples for a specific month/day dependent on, e.g., empirical variance of a criterion of interest. Another natural future research direction would be to directly combine the technique in [26] with ours.

B. Wind and Demand Distributions

Here we provide details on the models used for the stochastic wind and demand components, along with the data and test-cases they are based on.

1) Wind power distribution: Wind generation capacities are taken from [27], along with their daily mean profile. In addition, a monthly wind profile is adopted from [28]. The wind process mean $\mu_w(t)$ is obtained from the formula

$$\mu_w(t) = \mu_{w,\text{hourly}}(t_s) \cdot \mu_{w,\text{monthly}}(t_m),$$

where $\mu_{w,\text{hourly}}(t_s) \in \mathbb{R}_+^{n_+^{(g,w)}}$ is the daily wind mean profile at time-of-day t_s, and $\mu_{w,\text{monthly}}(t_m) \in [0,1]$ is the monthly wind profile relative to its peak at month t_m of the year; the latter is dictated according to the drawn process $\{y_m^{(t_m)}\}$, with distribution adopted from the data in [28].

The daily wind generation process W_{da} is multivariate normal:

$$W_{\text{da}} \sim \mathcal{N}\left(\mu_w(t), diag((p_{w,\sigma} \cdot \mu_w(t))^2)\right)$$

where $p_{w,\sigma} \in [0,1]$ is a constant ($= 0.15$) that multiplies the mean $\mu_w(t)$, to obtain a standard deviation that is a fixed fraction of the mean. $diag(x)$ is a square diagonal matrix, with the elements of x as its diagonal, assuming wind generators to be uncorrelated. W_{da} is truncated to stay in the range between 0 and the generator's capacity.

The hourly wind generation W_t is assumed to be a biased random walk, with expectation W_{da}; i.e., the real-time wind process is following the daily forecast up to some accumulated forecast error:

$$W_t = W_{\text{da}}(t) + \delta_t, \tag{6}$$

$$\delta_{t+1} = \delta_t + \epsilon_t, \tag{7}$$

where ϵ_t is Gaussian noise, $\epsilon_t \sim \mathcal{N}(0, 5 \cdot 10^{-3} \cdot W_{\text{da}}(0))$.

2) Load distribution: The daily load D_{da} is assumed to follow a distribution similar to the daily wind distribution W_{da}, with the same formula containing peak loads and daily profiles for each bus $\mu_{d,\text{hourly}}(t_s) \in \mathbb{R}_+^{n_b}$ with values taken from [27]. The fraction of mean for standard deviation is $p_{d,\sigma} = 0.02$.

Equivalently, the hourly load process D_t follows its day-ahead forecast D_{da} up to some accumulated error, as depicted in (6) and (7); in this case, the noise is $\epsilon_t \sim \mathcal{N}(0, 10^{-3} \cdot D_{\text{da}}(0))$.

VI. Machine Learning for a Short-Term Proxy

As mentioned earlier, in this work we utilize machine learning to build a short-term proxy. Thus, we replace exact solutions of the multiple UC problem instances, originating in (1e), with their quickly-predicted approximations. We use a well-known machine learning algorithm: *nearest neighbor classification* [29]; we thus call it UCNN. The methodology relies on a simple concept: creating a large and diverse data-set that contains samples of the environment and grid conditions along with their respective UC solutions. Then, during outage schedule assessment, instead of accurately solving the numerous UC problem instances, we simply choose the pre-computed UC solution with closest input conditions. Hence the phrase nearest neighbor. To confidently obtain high-quality approximate solutions, we generate the data-set so as to cover all relevant topologies that might be encountered during prediction. In our context, this implies a data-set that is $O(2^{|\mathcal{L}|})$, where $\mathcal{L}$ is the set of transmission lines for which outages ought to be performed. In general, this combinatorial dependence is not necessarily compulsory. Nevertheless, the question of accuracy degradation with smaller data-sets and more efficient data-set compositions are subject to future work.

The method's strength stems from the fact that during the optimization process (1), which is based on multiple outage schedule assessments, UC problem instances are often similar to previously computed ones. The initial data-set creation can either be done offline or online, i.e., by continually adding new solutions to the data-set as they become available. Nearest neighbour is attractive not only because of its ability to successfully handle the combinatorial problem structure (see [30]), but also because it seamlessly supports online updates. Being a "lazy" learner, its prediction capability is improved by simply adding accurately solved UC instances to its data-set. For the experiment described in this section, a data-set of 5000 UC problem instances was created in an 'offline' fashion. After obtaining this initial data-set, UCNN reduces computation time in several orders of magnitude, with relatively little compromise in quality [30]. The method is visualized in Fig. 4.

In addition to the direct UC approximation comparison in [30], we examine the resulting accuracy of outage scheduling

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2018.2889237, IEEE Transactions on Power Systems

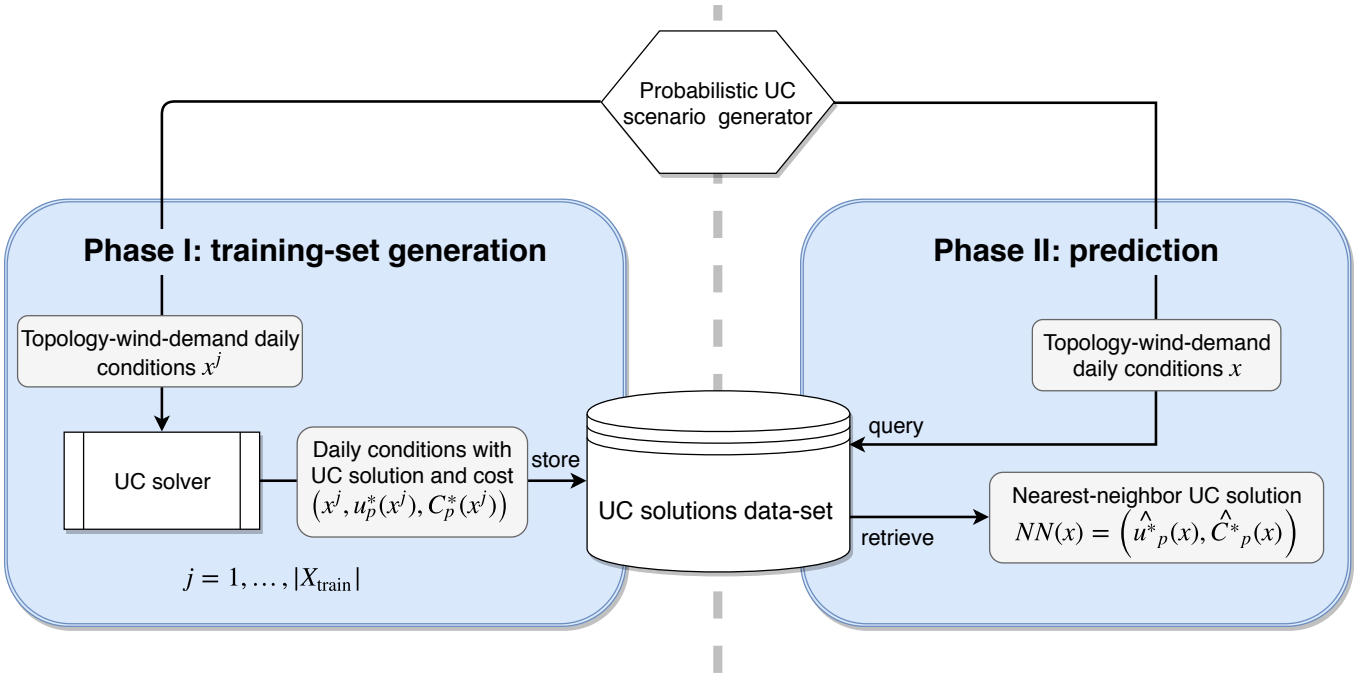


Fig. 4: UCNN algorithm diagram. In an initial phase, multiple UC inputs are generated, solved, and then stored along with their solutions and costs to create a diverse data-set, also referred to as *training set*. In a second phase, when a new UC problem instance is received, an approximate UC solution is obtained by finding a nearest-neighbor among the existing solutions in the data-set; i.e., a pre-solved similar problem instance. This is to replace the usage of the computationally expensive UC solver.

assessment when using UCNN to solve the short-term sub-problem instead of exact UC computations. To do so, we generate four arbitrary outage schedules under the configuration given in Section VIII. Then, for each of these schedules, we present in Fig. 5 means and standard deviations of several metrics in our simulation across the year's progress. These metrics are i) day-ahead operational costs and ii) LS amounts, taken from the short-term UC simulation. Additionally, they include the real-time values of iii) reliability as defined in (2) and vi) real-time operational costs. In all of these plots, the red curve is of an empty, no-outage schedule given as a baseline, evaluated using exact UC simulation; the blue and green curves are respectively based on exact UC and UCNN simulations of the arbitrary outage schedules. The persistent proximity of the green curve to the blue demonstrates the low approximation error when using UCNN, as it propagates to the four inspected metrics during the simulation.

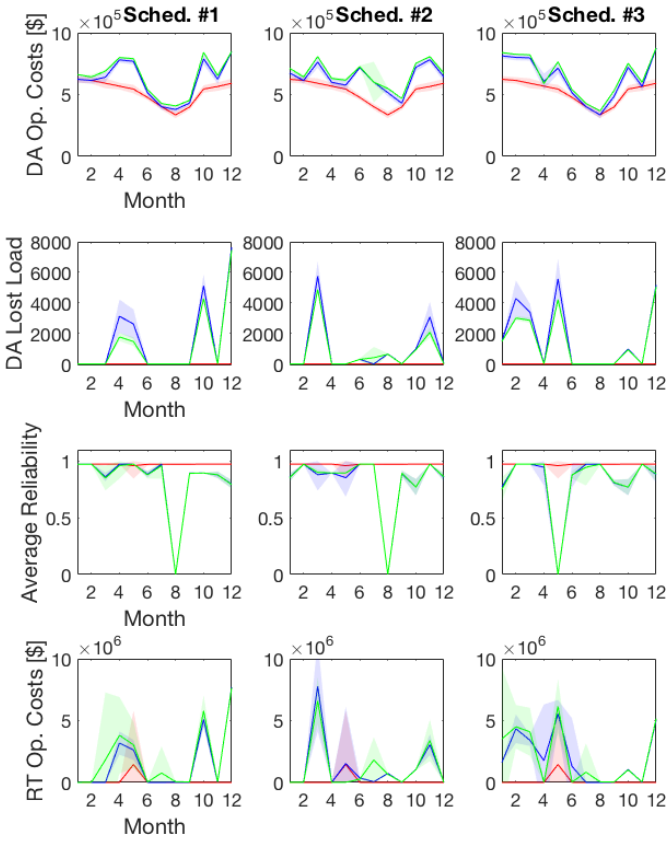


Fig. 5: Low proxy approximation errors demonstrated for IEEE-RTS79. Plotted are monthly costs of no-outage schedule as a reference (red) and four arbitrary outage schedules, evaluated with exact UC (blue) and UCNN (green).

VII. Distributed Cross Entropy Optimization

Problem (1) is a non-convex combinatorial stochastic optimization program with inner MILPs. Continuing the discussion in Section II-C, it is too complex for the objective and constraints to be expressed in explicit analytic form and for the program to be solved using gradient-based optimization. Furthermore, gradient-based approaches would preclude the option of utilizing smaller-horizon machine learning proxies such as our UCNN. For this reason, we choose a gradient-free simulation-based optimization approach. It is performed with distributed Monte-Carlo sampling, where multiple solutions in $\mathcal{U}_m$ are being assessed in parallel on multiple servers.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2018.2889237, IEEE Transactions on Power Systems

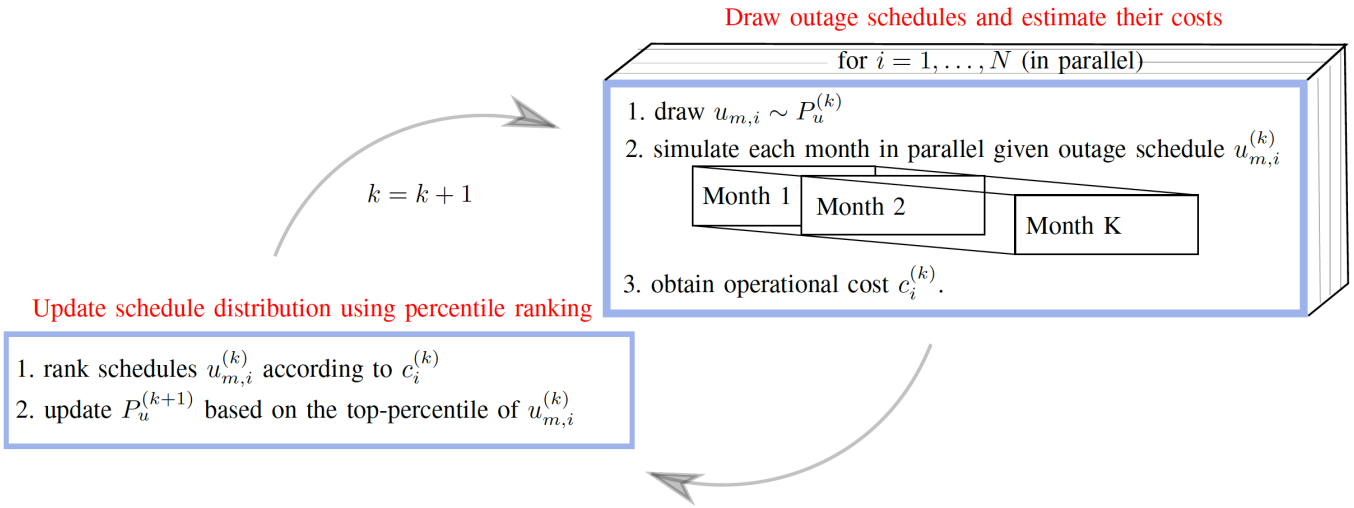


Fig. 6: A visualization of the Cross Entropy method.

Each month of such solution assessment is itself simulated in parallel.

In CE, a parametric distribution P_u is maintained over the solution space $\mathcal{U}_m$. Per each iteration k until convergence of $P_u^{(k)}$ to some final value, CE performs consecutive steps of:

1) Drawing N candidate outage schedules $u_{m,i}^{(k)} \sim P_u^{(k)}$, $i = 1, \ldots, N$. Then, evaluating their respective costs $c_i^{(k)}$ in parallel, based on simulated sampled scenario set $\hat{\mathcal{Z}}^{(k)}$ (generated with our Bayesian hierarchical window scenario sampler):

$$c_i^{(k)} = \frac{1}{|\hat{\mathcal{Z}}^{(k)}|} \sum_{\hat{Z} \in \hat{\mathcal{Z}}^{(k)}} \sum_{S_t \in \hat{Z}} C(S_t, u_{m,i}^{(k)}, u_s^*, u_{\text{RT}}^*).$$

2) Updating the parametric distribution of solutions $P_u^{(k)}$ based on the lowest (cheapest) 0.15-percentile of $c_i^{(k)}$.

This iterative process is visualized in Fig. 6.

In our simulations the outage scheduling horizon is one year with monthly candidate outage moments. Therefore, u_m is a binary matrix; i.e., $u_m \in \mathcal{U}_m = \{0,1\}^{|\mathcal{L}| \times 12}$. Entry $[u_m]_{\ell,m} = 1$ indicates a scheduled outage of line ℓ during month m.

We thus represent the CE distribution $P_u^{(k)}$ with a matrix of size $|\mathcal{L}| \times 12$ whose entries $[P_u^{(k)}]_{\ell,m} \in [0,1]$ depict outage likelihood. At iteration $k = 0$, these are all initialized to 0.5. As explained in the experiments section, according to the outage lists for IEEE-RTS79 and IEEE-RTS96 we need to schedule either 1 or 2 outages per each line in $\mathcal{L}$, depending on the line. Thus, per each row $[P_u^{(k)}]_\ell \in [0,1]^{12}$ (corresponding to line l) we respectively draw 1 or 2 entries out of the 12 candidate entries. This per-row sampling is performed by drawing one out of the $\binom{12}{1}$ or alternatively $\binom{12}{2}$ entry combinations based on their proportional probability, calculated using matrix $P_u^{(k)}$. The first step of the CE algorithm is thus to iterate the above procedure N times for sampling $u_{m,i}^{(k)}$, $i = 1, \ldots, N$.

The second step of the CE algorithm is to update $P_u^{(k)}$ as follows. Let $\mathcal{I}_c \subset \{1, \ldots, N\}$ be the set of indices of the lowest 0.15-percentile of costs $c_i^{(k)}$. Then, we update the entries $P_u^{(k+1)} = \frac{1}{|\mathcal{I}_c|} \sum_{j \in \mathcal{I}_c} u_{m,j}^{(k)}$; i.e., the entries of $P_u^{(k+1)}$ are set to be the average of the top solutions' entries.

Lastly, our criterion for convergence is when the entropy of $P_u^{(k+1)}$ drops below some small $\epsilon > 0$. This occurs when all entries are sufficiently close to either 0 or 1.

As for constraint satisfaction, it is ensured in the following way: Chance-Constraints (1b)-(1c) are evaluated empirically and their violation is penalized with increasing-slope barrier functions; Feasibility Constraint (1d) is ensured via the structure of the CE parametric distribution described above; Inner Constraints (1e)-(1f) are ensured via the solvers used for simulating them.

VIII. Simulation Studies

We run our simulations on a Sun cluster with Intel Xeon(R) CPUs @2.53GHz, containing a total of 300 cores, each with 2GB of memory. All code is written in Matlab [31]. In each iteration of the CE algorithm, we assess the objective and constraint values of 75 drawn outage schedules in parallel, while also parallelizing the simulation of each of the 12 months. The simulation parameters introduced in Section V, depicting daily and hourly trajectory length and multiplicity, are $T_s = 3$, $N_s = 4$, $T_{\text{RT}} = 24$, $N_{\text{RT}} = 2$. This totals a year-long trajectory which is sampled 3 times.

A. Test-Cases and Outages

In our simulation, we consider the IEEE-RTS79 and IEEE-RTS96 test-cases [32]. We adopt updated generator parameters from Kirschen et al. [33], namely their capacities, min-output, ramp up/down limits, min up/down times, price curves and start-up costs. Peak loads and hourly demand means are based on data from the US published in [27]. Capacities and means of hourly wind generation are also based on real data, taken from [27]. Value of lost load is set to VOLL $= 1000[\frac{\$}{MWh}]$, taken from [34] and wind-curtailment price is set to $C_{\text{WC}} = 100[\frac{\$}{MWh}]$, taken from [35].

Additionally, we slightly modify the test-cases so as to create several 'bottleneck' areas to provide conditions for

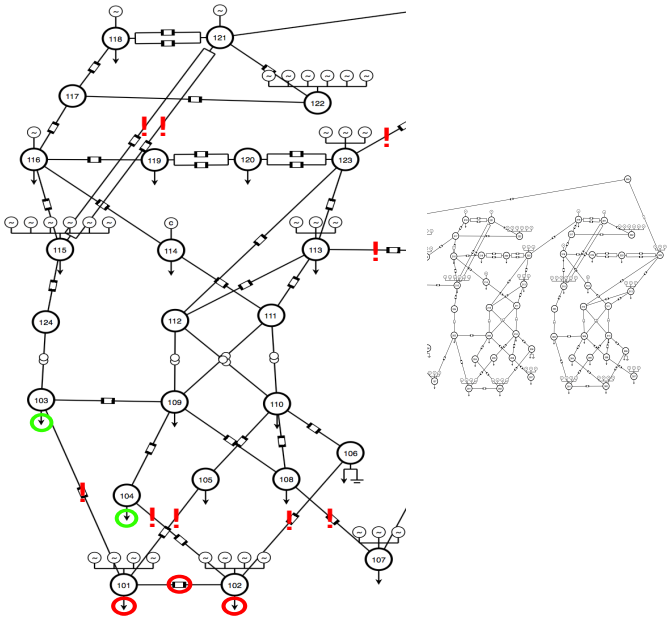
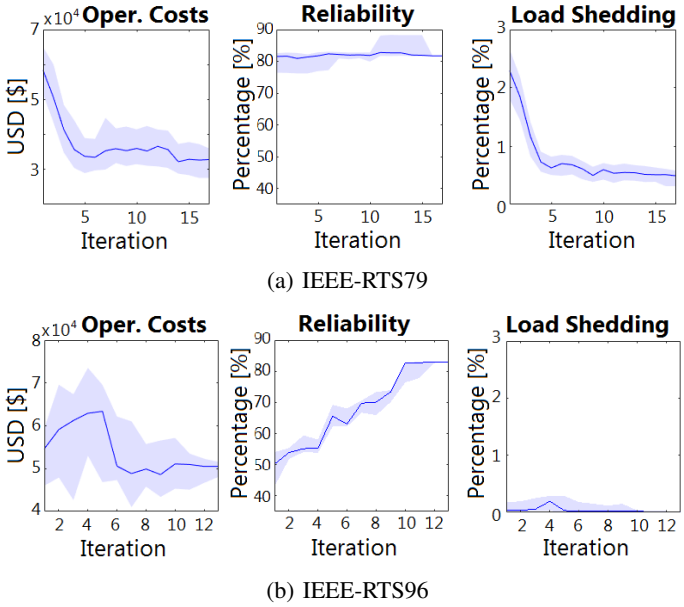


Fig. 7: Modifications and target outages in the IEEE-RTS96 test-case (conducted per each of the zones, though plotted on one for simplicity). Red circles denote removal, green circles denote increase, and red exclamation marks denote candidate planned outage. In RTS79, the same modifications and outages are conducted in its single zone (except of outages in interconnections since they do not exist there).



(a) IEEE-RTS79

(b) IEEE-RTS96

Fig. 8: Convergence of the Cross Entropy method. Plotted are medians with upper and lower quartiles of three metrics for the top CE percentile: operational costs (redispatch, wind curtailment and unit re-commitment), average reliability, and average load shedding .

variant outage schedule qualities. In RTS79, these modifications include i) removal of transmission line between bus 1 and 2, and ii) shift of loads from buses 1 and 2 to buses 3 and 4, respectively. In the case of RTS96, the same exact modifications are replicated to all three zones.

Next, we specify the outage lists. For RTS79, it is composed of 13 outages: 2 outages per each of lines $\{2, 3, 4, 5, 25, 26\}$ and 1 outage for line 11. For RTS96, the list is composed of 30 outages: 9 per each zone plus 3 for the interconnections. Specifically, in the first zone of RTS96 we have 2 outages per each of lines $\{2, 3, 4, 5\}$ and 1 outage for line 11; these are replicated similarly to the equivalent lines in the second and third zones. The additional interconnection outages are 1 per each of lines $\{12, 119, 120\}$. The test-case modifications and outages are visualized in Fig. 7.

B. UCNN Data-Set Size Complexity

We now briefly relate to the UCNN data-set size and, for clarity, distinguish it from the optimization search space size. The latter is, in general, much larger than the former. To see this, let us denote by K the number of outages required per each line in the outage list $\mathcal{L}$. Then each of the $|\mathcal{L}|$ lines has $\binom{|\mathcal{T}_m|}{K} = O(|\mathcal{T}_m|^K)$ possible outage allocations throughout the planning horizon, resulting in an optimization search space sized $O(|\mathcal{T}_m|^{K|\mathcal{L}|})$. Contrarily, as explained in Section VI, the required UCNN data-set size is proportional to the outage combinations, i.e., $O(2^{|\mathcal{L}|})$; it does not depend on $\mathcal{T}_m$ or K.

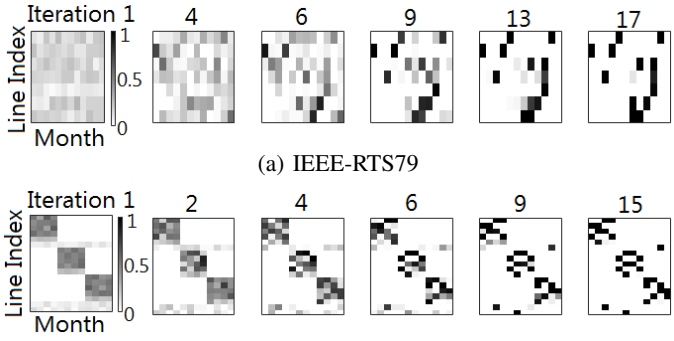
Nevertheless, the exponential growth in $\mathcal{L}$ poses a scalability issue when generating the UCNN data-set. Given our setup, as specified in Subsection VIII-A, the resulting size for RTS-96 is $O(2^{3\times5+3})$. To mitigate this, when scheduling outages for RTS96, we assume that the year is partitioned into three periods of 4 months, and each of the three "zone operators" is exclusively allocated with distinct 4 months to conduct her 9 outages. This is enforced via the feasibility constraint (1d). As for the outages of the additional 3 interconnections, those are independent and free to be chosen to any of the year's 12 months. We thereby do away with the exponential dependence of UCNN's data-set complexity in the number of zones, i.e., reduce the $O(2^{3\times5+3})$ training set size to $O(3 \times 2^{5+3})$.

C. Results

Fig. 8 exhibits the fast convergence as expected from CE when solving (1) for the two test-cases, along with intriguing differences between them. It gives the median with upper and lower quartiles of the top CE percentile for three metrics: operational costs from (1a) (redispatch, wind curtailment and unit re-commitment), average reliability from (1b), and average LS from (1c). In both test-cases, operational costs significantly drop. As for the reliability and LS, a somewhat complementary behavior is observed for the two test-cases. The reliability in RTS79 starts off with high enough values, 83%, to satisfy its constraint (1b), while the LS amount starts high and quickly drops to a satisfying level of 0.4%. The exact opposite happens for RTS96: reliability starts low and increases drastically to 83%, while LS values consistently remain low throughout the optimization process, stabilizing at 0.05%.

We also visualize the convergence in the space of outage schedules in Fig. 9 via gray-level-mapped matrices. The rows of these matrices denote the transmission lines and their

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2018.2889237, IEEE Transactions on Power Systems

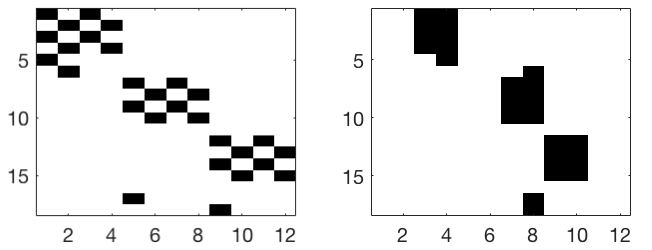


(a) IEEE-RTS79

(b) IEEE-RTS96

Fig. 9: A visualization of drawn outage schedules throughout selected iterations of the CE optimization process, demonstrating convergence to a single schedule. Each iteration is represented with a gray-level-mapped matrix, whose rows denote the transmission line index out of those undergoing outages and their columns denote the outage moments. For a given entry, the gray-level corresponds to the relative intensity of outages selected for the specific line-month combination.

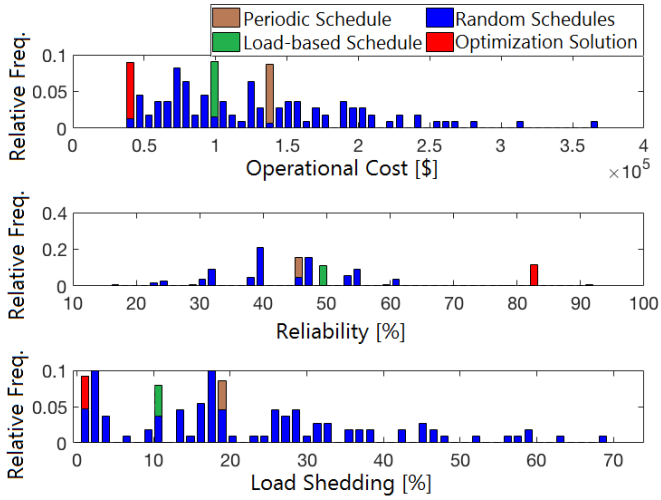
columns denote the outage moments. For a given entry, the gray-level corresponds to the relative intensity of outages selected for the specific line-month combination. The initial CE iteration begins with uniformly-drawn candidate outages moments, followed by convergence towards a single solution. In the case of RTS96, the zonal time-allocation can be seen in the form of three shaded blocks of entries, with the three interconnection outages in the form of three shaded independent rows.



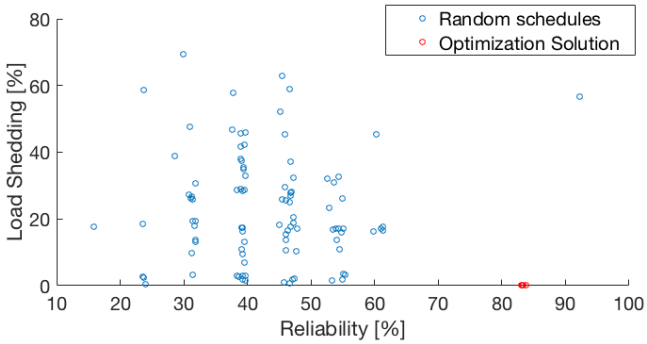
(a) Periodic outage schedule (b) Load-based outage schedule

Fig. 10: Two heuristic outage schedules that experts would possibly consider. For explanation on the visualization method see Fig. 2.

To demonstrate the efficacy of our optimized outage schedule, we compared it with two heuristics that mimic possible expert outage schedules, as well as multiple arbitrary outage schedules. The first expert heuristic, visualized in Fig. 10(a), performs periodic maintenance; this method is presently used by various European TSOs [36]. The second expert heuristic, visualized in Fig. 10(b), schedules as many of the outages during times when the load is low; we chose it since it was found to be optimal in [10]. Notice that these two expert plans are confined to our exclusive zonal time separation, due to the setup described in Section VIII-B. Per each of those heuristics, as well as for our optimization solution, we ran 10



(a) Histograms of operational cost, reliability, and load-shedding.

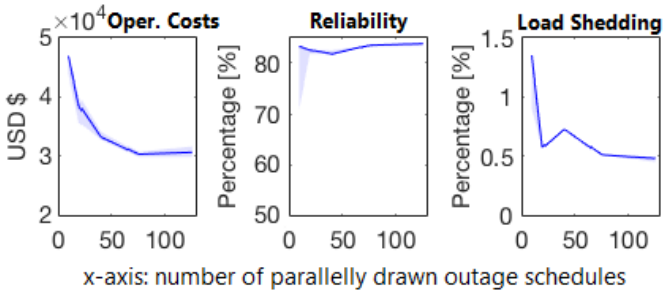


(b) Reliability vs. load-shedding scatter plot

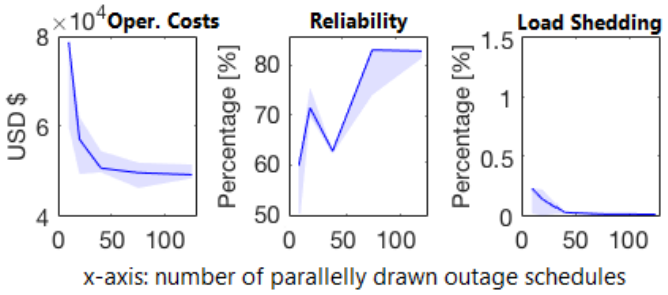
Fig. 11: A comparison on IEEE-RTS96 of 100 arbitrary outage schedules to 10 instances per each of the heuristics depicted in Fig. 10, and our optimization solution. Altogether there are 130 instances of scenario-evaluation runs, normalized in the relative frequency histograms in (a). In (a), the dominance of our solution is shown in all three inspected metrics. There is one single exception to its optimality: a random schedule with $> 90\%$ reliability. However, in (b), where the reliability vs. load-shedding tradeoff is depicted, this single schedule is shown to suffer from high load shedding.

evaluations. Lastly, we evaluated 100 random schedules that comply with the zonal time allocation. Fig. 11(a) displays operational cost, reliability, and LS histograms of the 130 evaluated schedules. The load-based schedule outperforms the periodic one. However, both are outperformed by optimization solution, which consistently exhibits the lowest operational costs, highest reliability, and lowest LS. One exception to its optimality is a single random schedule that achieves reliability greater than 90%. To further examine it, we added a scatter plot in Fig. 11(b) to capture the reliability vs. LS tradeoff. It reveals that the aforementioned highly-reliable random schedule suffers from high load-shedding values, as opposed to our optimization solution.

In our last simulation we tested the quality of the optimization solution as a function of the number of drawn outage



(a) IEEE-RTS79



(b) IEEE-RTS96

Fig. 12: Quality of the optimization solution as a function of the number of outage schedules drawn per each iteration of Cross Entropy. The figure depicts the tradeoff between computational cost and reachable objective-value/constraint-satisfiability. Plotted are medians with upper and lower quartiles of three metrics: operational costs (redispatch, wind curtailment and unit re-commitment), average reliability, and average load shedding. In both test-cases, the constraints are satisfied and the objective saturates at $N = 75$ drawn outages per iteration, in terms of median performance.

schedules per each iteration of CE. Namely, we ran the CE optimization procedure until convergence (for the convergence criterion see Section VII) 5 times, with N (number drawn schedules per CE iteration) taken from $[10, 20, 40, 75, 125]$. Fig. 12 gives the converged values in terms of operational cost, reliability and LS. The plots reveal that for both test-cases $N = 75$ is sufficient to satisfy the reliability and LS constraints, which respectively should be above 80% and below 0.5% (see Section III-B). Also, no significant reduction in operational costs is witnessed beyond beyond that value.

IX. Conclusion

The power system infrastructure is ageing, and its maintenance is becoming more and more costly and complex. This calls for new sophisticated outage scheduling tools, that will handle uncertainty and coordination with operations. The scenario assessment framework introduced in this work enables detailed evaluation of implicit intricate implications an outage schedule inflicts on a power system. We harness the power of machine learning and distributed computing to tractably perform multiple schedule assessments in parallel. We also wrap it in an optimization framework that finds convincingly high-quality schedules. An additional, straightforward application of the methodologies introduced here is assessment of a predefined maintenance schedule considered by experts.

The focus of this work is in the probabilistic framework and hierarchical methodologies. Nevertheless, we believe it enables gaining new insights for both academic networks and more complex real-world test-cases. The proposed framework is flexible and can be adapted to different practical cost functions and reliability criteria. But most importantly, it is completely data-driven – it does not rely on knowing the probabilities of scenarios; it solely requires access to a black-box scenario generator. This conforms with resampling real-life data.

Lastly, this work raises the question of the benefit of real-time simulation for planning purposes compared to more classical methods. To answer that question, one should first understand how to evaluate and compare classical methods to data-driven methods in terms of known metrics, given the inherent differences between them listed in Section II-C. In this work, we attempted to bridge some of the gaps between the two schools, by focusing on metrics which we found comparable. Other metrics, such as the probabilistic N-1 reliability criterion, required a certain level of flexibility in order to become "data-compatible". Nonetheless, the authors encourage future research on devising methods for performance comparison both in simulation and real systems.

X. Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No 608540, project acronym GARPUR. The scientific responsibility rests with the authors.

References

[1] GARPUR Consortium, "D5.1, Functional analysis of asset management processes," http://www.garpur-project.eu/deliverables, Oct. 2015.

[2] G. Dalal, E. Gilboa, and S. Mannor, "Distributed scenario-based optimization for asset management in a hierarchical decision making environment," in *Power Systems Computation Conference*, 2016.

[3] E. Karangelos and L. Wehenkel, "Probabilistic reliability management approach and criteria for power system real-time operation," in *Power Systems Computation Conference*, 2016.

[4] GARPUR Consortium, "D2.2, Guidelines for implementing the new reliability assessment and optimization methodology," http://www.garpur-project.eu/deliverables, Oct. 2016.

[5] ——, "D5.2, Pathways for mid-term and long-term asset management," http://www.garpur-project.eu/deliverables, Oct. 2016.

[6] A. Abiri-Jahromi, M. Parvania, F. Bouffard, and M. Fotuhi-Firuzabad, "A two-stage framework for power transformer asset maintenance management – Part I: Models and formulations," *Power Systems, IEEE Transactions on*, vol. 28, no. 2, pp. 1395–1403, 2013.

[7] Y. Jiang, Z. Zhong, J. McCalley, and T. Voorhis, "Risk-based maintenance optimization for transmission equipment," in *Proc. of 12th Annual Substations Equipment Diagnostics Conference*, 2004.

[8] M. Marin, E. Karangelos, and L. Wehenkel, "A computational model of mid-term outage scheduling for long-term system studies," in *PowerTech, 2017 IEEE Manchester*. IEEE, 2017, pp. 1–7.

[9] Y. Fu, M. Shahidehpour, and Z. Li, "Security-constrained optimal coordination of generation and transmission maintenance outage scheduling," *IEEE Trans. on Power Systems*, vol. 22, no. 3, pp. 1302–1313, 2007.

[10] L. Wu, M. Shahidehpour, and Y. Fu, "Security-constrained generation and transmission outage scheduling with uncertainties," *IEEE Transactions on Power Systems*, vol. 25, no. 3, pp. 1674–1685, 2010.

[11] H. Pandzic, A. J. Conejo, I. Kuzle, and E. Caro, "Yearly maintenance scheduling of transmission lines within a market environment," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 407–415, 2012.
[12] C. D. López, B. Idlbi, T. Stetz, and M. Braun, *Shortening Quasi-Static Time-Series Simulations for Cost-Benefit Analysis of Low Voltage Network Operation with Photovoltaic Feed-In*. Universitätsbibliothek Dortmund, 2015.
[13] R. Canyasse and G. Dalal, "Supervised learning for optimal power flow as a real-time proxy," in *IEEE PES Innovative Smart Grid Technologies (ISGT 2017)*, vol. 8, 2017.
[14] Y. Ng, S. Misra, L. A. Roald, and S. Backhaus, "Statistical learning for DC optimal power flow," in *Power Systems Computation Conference*, 2018.
[15] L. Duchesne, E. Karangelos, and L. Wehenkel, "Using machine learning to enable probabilistic reliability assessment in operation planning," in *Power Systems Computation Conference*, 2018.
[16] I. M. Cecílio, J. R. Ottewill, J. Pretlove, and N. F. Thornhill, "Nearest neighbors method for detecting transient disturbances in process and electromechanical systems," *Journal of Process Control*, vol. 24, no. 9, pp. 1382–1393, 2014.
[17] A. A. Majd, H. Samet, and T. Ghanbari, "k-NN based fault detection and classification methods for power transmission systems," *Protection and Control of Modern Power Systems*, vol. 2, no. 1, p. 32, 2017.
[18] H. Everett III, "Generalized lagrange multiplier method for solving problems of optimum allocation of resources," *Operations research*, vol. 11, no. 3, pp. 399–417, 1963.
[19] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische mathematik*, vol. 4, no. 1, pp. 238–252, 1962.
[20] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.
[21] J. Lfberg, "YALMIP : A toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. [Online]. Available: http://users.isy.liu.se/johanl/yalmip
[22] "IBM ILOG CPLEX Optimizer," http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/, 2010.
[23] R. S. Dembo, "Scenario optimization," *Annals of Operations Research*, vol. 30, no. 1, pp. 63–80, 1991.
[24] S. Pineda, J. M. Morales, and T. K. Boomsma, "Impact of forecast errors on expansion planning of power systems with a renewables target," *European Journal of Operational Research*, 2016.
[25] B. A. Mather, "Quasi-static time-series test feeder for PV integration analysis on distribution systems," in *Power and Energy Society General Meeting, 2012 IEEE*. IEEE, 2012, pp. 1–8.
[26] B. Mather, "Fast determination of distribution-connected pv impacts using a variable time-step quasi-static time-series approach: Preprint," National Renewable Energy Laboratory (NREL), Golden, CO (United States), Tech. Rep., 2017.
[27] E. E. D. University of Washington, "Renewable energy analysis lab," http://www.ee.washington.edu/research/real/library.html.
[28] A. Ouammi, H. Dagdougui, R. Sacile, and A. Mimet, "Monthly and seasonal assessment of wind energy characteristics at four monitored locations in Liguria region (Italy)," *Renewable and Sustainable Energy Reviews*, vol. 14, no. 7, pp. 1959–1968, 2010.
[29] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.
[30] G. Dalal, E. Gilboa, S. Mannor, and L. Wehenkel, "Unit commitment using nearest neighbor as a short-term proxy," in *Power Systems Computation Conference*, 2018.
[31] Matlab, "The language of technical computing," http://www.matlab.com.
[32] C. Grigg, P. Wong, P. Albrecht, R. Allan, M. Bhavaraju, R. Billinton, Q. Chen, C. Fong, S. Haddad, S. Kuruganty *et al.*, "The IEEE reliability test system-1996. A report prepared by the reliability test system task force of the application of probability methods subcommittee," *IEEE Transactions on power systems*, vol. 14, no. 3, pp. 1010–1020, 1999.
[33] H. Pandzic, T. Qiu, and D. S. Kirschen, "Comparison of state-of-the-art transmission constrained unit commitment formulations," in *Power and Energy Society General Meeting (PES), 2013 IEEE*, 2013.
[34] Y. Dvorkin, H. Pandzic, M. Ortega-Vazquez, D. S. Kirschen *et al.*, "A hybrid stochastic/interval approach to transmission-constrained unit commitment," *IEEE Transactions on Power Systems*, vol. 30, no. 2, pp. 621–631, 2015.
[35] R. Loisel, A. Mercier, C. Gatzen, N. Elms, and H. Petric, "Valuation framework for large scale electricity storage in a case with wind curtailment," *Energy Policy*, vol. 38, no. 11, pp. 7323–7337, 2010.
[36] GARPUR, "Functional analysis of reliability management," 7th framework programme, EU Commission grant agreement 608540, Sep. 2014. [Online]. Available: https://project.sintef.no/eRoom/energy2/GARPUR-FP7proposal/0_1489e

Gal Dalal graduated in Electrical Engineering in 2013 and received his Ph.D. degree in 2018, both from Technion, Israel Institute of Technology. He also worked as a Machine Learning Research Scientist for IBM Research and Google DeepMind. His research interests lie on the intersection between power systems and probabilistic optimization methods, machine learning and reinforcement learning.

Elad Gilboa received a B.A in electrical and computer engineering and M.E. in biomedical engineering respectively in 2004 and 2009, from Technion, Israel, and a PhD in electrical engineering in 2014 from Washington University in St. Louis, USA. After which he did postdoctoral work in the Machine Learning Group at the Technion. His research includes machine learning and statistical single processing, with focus on making these methods scalable to real world problems.

Shie Mannor received the B.Sc. degree in electrical engineering, the B.A. degree in mathematics, and the Ph.D. degree in electrical engineering from the Technion-Israel Institute of Technology, Haifa, Israel, in 1996, 1996, and 2002, respectively. From 2002 to 2004, he was a Fulbright scholar and a postdoctoral associate at M.I.T. He was with the Department of Electrical and Computer Engineering at McGill University from 2004 to 2010 where he was the Canada Research chair in Machine Learning. He has been with the Faculty of Electrical Engineering at the Technion since 2008 where he is currently a professor. His research interests include machine learning and pattern recognition, planning and control, multi-agent systems, and communications.

Louis Wehenkel graduated in Electrical Engineering (Electronics) in 1986 and received the Ph.D. degree in 1990, both from the University of Liège, where he is full Professor of Electrical Engineering and Computer Science. His research interests lie in the fields of stochastic methods for systems control, optimization, machine learning and data mining, with applications in complex systems, in particular large scale power systems planning, operation and control, industrial process control, bioinformatics and computer vision.