Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

ACCELERATING SPLITTING ALGORITHMS FOR POWER GRID RELIABILITY ESTIMATION

Wander S. Wadman

Smarter Energy and Environmental Sciences
IBM Research
Yorktown Heights, NY 10598 USA

Mark S. Squillante
Soumyadip Ghosh

Mathematical Sciences
IBM Research
Yorktown Heights, NY 10598 USA

ABSTRACT

With increasing penetration of intermittent power generation sources like wind and solar farms, overload risks in power grids due to imbalances in supply and demand of energy has become a serious concern. We model the flow of electricity through a power grid as a functional transformation of a multidimensional Ornstein-Uhlenbeck process of renewable energy injection. Previously, a rare event simulation technique called splitting, based on large deviations results has been proposed as the risk assessment method. This method requires solving a nonlinear optimization problem for every time step in every generated sample path, so significant computational challenges remain in scaling to realistic networks. We propose a new algorithmic approach to implement the large deviations splitting method that derives and exploits fundamental properties of the rate functions in order to significantly speed up the pathwise optimizations. Experimental results show a significant reduction in effort compared to a conventional numerical approach.

1 INTRODUCTION

Power transmission grids have to handle increasing uncertainty as the proportion of renewables in the generation portfolio rises. Given dispatch levels for all scheduled generators, a deterministic mean forecast for all renewable generation, and the predicted mean demand information, the traditional deterministic power flow (PF) tool (Wood and Wollenberg 1996) solves the AC power flow equations to determine the voltage magnitudes at buses and the flows of power through the branches in the grid. Slack buses are used to hedge against any residual mismatch between the total load and generation. The aim of PF prediction is to check for congestion, which is defined as the event when the voltage at a bus or power flow through a transmission line exceeds specified capacity limits. This stability analysis provides system operators with an early-warning on emerging congestion conditions and possible voltage/thermal violations. Current practice deems it sufficient to run this step into the near future of 1-5 minutes. However, the substantial volatility of renewable generation and hence variability in its prediction requires a stochastic prediction to foresee potential congestion in the grid, as the share of renewables in the total generation portfolio rises. Renewable generation volatility affects both the available generation via large-scale renewable farms and the prediction of net load due to the increasingly wide-spread use of small behind-the-meter renewable generation in homes and local neighborhoods.

The literature on impact of renewables uncertainty has mostly concentrated on the day-ahead market (unit-commitment) and the economic dispatch or optimal power flow decision problems (Summers, Warrington, Morari, and Lygeros 2015), while we seek methods to predict the probability of congestion events on the transmission grid due to load and/or renewables uncertainty after the unit commitment and dispatch schedules are (mostly) fixed. A Monte Carlo simulation method for linearized power flow equations is developed in Leite da Silva and Arienti (1990) and Wadman, Bloemhof, Crommelin, and Frank (2012). Fast Fourier transform used in a convolution technique is introduced in Allan, Leite da Silva, and Burchett

978-1-5090-4486-3/16/$31.00 ©2016 IEEE

(1981) for linearized power flow models, the most common form of which assume a direct-current (DC) circuit model (see Section 2). Approximate point estimate methods with some assumptions on the random variables is presented in Su (2005). The authors of Zhang and Lee (2004) analytically calculate probabilistic distribution functions for power flows from theories of cumulants and Gram-Charlier expansion for linear DC power flow equations. The paper Esmaili, Amjady, and Shayanfar (2010) describes a single-period AC power flow congestion prediction scheme using a straightforward Monte Carlo sampling scheme.

These above-mentioned methods either model only a single period, or linearize the power flow equations or both. The state-of-the-art short-term demand and renewables forecasting tools (Alzate and Sinn 2013) can provide a distributional description of the possible outcomes into a longer horizon of the next 6-24 hours. In Wadman, Crommelin, and Zwart (2016), a multiperiod model is described for a linearized power flow model in order to predict the probability of congestion over a single critical line in the power grid. Power grids are well-designed and typically run well within their operational bounds. So, congestions are low-probability but extremely costly occurrences. To efficiently estimate these probabilities under a probabilistic renewable generation model, Wadman, Crommelin, and Zwart (2016) employ a variant of rare event Monte Carlo simulation called a splitting method (Section 3) to sample more often from multiperiod sample paths that lead to congestion events. This method is particularly suited to the multiperiod PF formulation because the path-dependent evolution of the multiperiod system is a challenge to most efficient sampling schemes, but is a natural fit to the method of splitting.

Crucial to an efficient implementation of splitting is the choice of a parameter called the importance function. Wadman, Crommelin, and Zwart (2016) show that the splitting method can be asymptotically efficient (in a certain precise sense) in estimating the small probability if the importance function is defined in terms of a large deviations based rate function derived for the congestion event (see Section 3). Computing this optimal rate function can be computationally challenging: as seen in Section 3, the optimal rate function is sample path dependent, and estimating it requires solving a state-dependent one-dimensional nonlinear optimization problem at every period of generating a single sample path in order to determine the most likely time of failure into the future up to the horizon given the current state of the grid. In Wadman, Crommelin, and Zwart (2016), these individual optimization problems are solved by enumerating the objective function over the decision space, which limits the efficacy of the method in handling large power grids.

We propose a novel algorithmic approach to accelerate the solving of these path-dependent optimization problems. This is achieved by analyzing the gradient of the objective function, which can be written in a certain form (see Section 4) that suggests a natural bisecting algorithm to pare down the feasible set iteratively to quickly hone in on the region where the optimal solution may lie. The resulting algorithm, called the Tree-based Bisection Algorithm (TBA) is described in Section 5. Extensive numerical experiments were conducted on a standard IEEE example of a power grid (Christie 2006), and results of which (in Section 6) show a remarkable saving in the function evaluations required in the TBA against the existing method from Wadman, Crommelin, and Zwart (2016). In most cases, TBA seems to require an order of magnitude smaller number of function evaluations than the conventional method requires.

2 POWER GRID MODEL

Let the vector-valued stochastic process $\{X^\varepsilon(t), t \geq 0\}$ denote n uncertain power injections of a power grid as a function of time t. We define this process as the multidimensional Ornstein-Uhlenbeck (OU) process

$$dX^\varepsilon(t) = D(\mu - X^\varepsilon(t))dt + \sqrt{\varepsilon} L dW(t), \quad X^\varepsilon(0) = x_0, \tag{1}$$

where $D := \text{diag}(\theta_1, \ldots, \theta_n) \in \mathbb{R}^{n\times n}$ is a diagonal matrix with mean-reverting terms $\theta_1, \ldots, \theta_n > 0$ on the diagonal. Here, $\mu \in \mathbb{R}^n$ is the vector of long-run means, $\varepsilon > 0$ is a scalar called the rarity parameter, $L \in \mathbb{R}^{n\times n}$ is a lower triangular matrix with $\Sigma := LL^\top$ the covariance matrix of $LW(1)$, and $W(t)$ is a vector of i.i.d. standard Brownian motions. Then $X_i^\varepsilon(t)$ is clearly a one-dimensional OU process with mean-reverting term θ_i, long-term mean μ_i, volatility $\sqrt{\varepsilon \Sigma_{ii}}$ and initial value $x_{0,i}$. The injection pattern at node i therefore deviates according to $\sqrt{\varepsilon \Sigma_{ii}}$ but reverts back to mean μ_i with force θ_i. Dependencies between different power injections are captured in the model through L, thus reflecting the correlation between the meteorological sources of renewable energy or between consumption at different nodes. Wadman, Crommelin, and Zwart (2016) show that $X^{\varepsilon}(t)$ follows a multivariate normal distribution.

We define the function $p : \mathbb{R}^n \to \mathbb{R}$ that maps the power injections to the power flowing through a specific grid connection. A common choice for p involves alternating current (AC) power flow equations, which compose a nonlinear algebraic system of steady state equations that relates the power injections at each grid node to the voltages at all nodes. This nonlinear algebraic system has to be solved numerically for the nodal voltages in order to compute a connection power flow at some time t given the power injections at that time. Ohm's law and the definition of power will then immediately yield the power flow through a connection. We refer to (Grainger and Stevenson 1994, Chapter 9) for additional details.

Another choice for p is a linear function of the power injections to the power flow through the connection of interest, namely

$$p(x) = v^{\top} x \tag{2}$$

for some constant vector $v \in \mathbb{R}^n$. The direct current (DC) power flow equations form a well-known example of this choice for p (Seifi and Sepasian 2011, Appendix A). Moreover, linear functions for p have been derived for radial AC networks (Low 2014). In this paper, we assume that p is a deterministic and continuous linear function of x that solves a system of steady state equations. Additional details on this linear model can be found in (Wadman, Crommelin, and Zwart 2016).

Our specific interest lies in the overload probability

$$\gamma := \mathbb{P}\left\{ \sup_{\tau \in (0,T]} p(X^{\varepsilon}(\tau)) \geq P_{\max} \right\}$$

before some time $T > 0$, where $P_{\max} > 0$ is the maximum allowed value of power flowing through the connection. Since X^{ε} and p are continuous, the event of interest will occur precisely when the power flow equals $P_{\max}$ at least once before T, so we have

$$\gamma = \mathbb{P}\left\{ \exists\ \tau \in (0,T] : p(X^{\varepsilon}(\tau)) = P_{\max} \right\}. \tag{3}$$

We will now show that if no overload occurs under nominal loading conditions, probability γ is small for vanishing ε (explaining the term rarity parameter). Suppose that no overload occurs if the power injections are equal to their expectation:

$$p(\mathbb{E}[X^{\varepsilon}(t)]) < P_{\max} \text{ for all times } t. \tag{4}$$

Wadman, Crommelin, and Zwart (2016) show that

$$\mathbb{E}[X^{\varepsilon}(t)] = e^{-Dt}(x_0 - \mu) + \mu$$

where the exponential is a matrix exponential, and thus assumption (4) implies that there is neither an overload at the starting time, $p(x_0) < P_{\max}$, nor under average circumstances, $p(\mu) < P_{\max}$, nor under the most likely circumstances in between. Hence, for vanishing ε, $\mathbb{P}\{p(X^{\varepsilon}(t)) \geq P_{\max}\}$ goes to zero for all t. We then can conclude that, for fixed T, γ goes to zero as ε vanishes, which is why γ is a rare event probability if ε is small.

From large deviations theory, we know that the most likely path to the rare event set will become dominant as the rare event probability vanishes, i.e., the decay rate of the vanishing probability converges to the good rate function of this most likely path. In particular, the minimum good rate function of the probability in (3) converges to $I^*(0,x_0)$, suggesting the following approximation for small ε:

$$\gamma \approx e^{-I^*(0,x_0)/\varepsilon}.$$

This approximation serves to help distinguish connections with a significant overload probability, where the most likely path sheds light on the typical combination of power injection paths that leads to an overload. We will next use this approximation to construct a suitable importance function for a splitting simulation, expressing the decay rate $I^*(s,x_s)$ as a minimization over $\tau \in (s,T]$ of an infimum $g_{s,x_s}(\tau)$ for general τ:

$$I^*(s,x_s) = \inf_{\tau\in(s,T]} g_{s,x_s}(\tau), \qquad \text{with} \qquad g_{s,x_s}(\tau) := \inf_{x:p(x(\tau))=P_{\max}} I_{s,x_s}(x). \tag{5}$$

3 THE SPLITTING METHOD

Our goal is to estimate the overload probability γ in (3) using crude Monte Carlo (CMC) simulation, where following Wadman, Crommelin, and Zwart (2016) we sample trajectories from the discretization X_t^ε of $X^\varepsilon(t)$ in (1) to see whether $p(X_t^\varepsilon) > P_{\max}$ at some discrete time t. The CMC estimator for γ, defined as

$$\hat{\gamma}_{\mathrm{CMC}} := \frac{1}{N}\sum_{i=1}^{N} 1_{\{\exists\ \tau\in(0,T]:p(X_\tau^\varepsilon)\geq P_{\max}\ \text{in sample}\ i\}},$$

is in principle biased because rare event occurrences between subsequent time steps are ignored. However, consistent with (Wadman, Crommelin, and Zwart 2016), we shall assume that the discrete time steps are chosen sufficiently small to ignore this discretization bias.

One may need a prohibitively large number of samples to estimate a very small probability for such rare events using CMC simulation, since the squared relative error of the CMC estimator is given by

$$\frac{\operatorname{Var}\hat{\gamma}_{\mathrm{CMC}}}{\gamma^2} = \frac{\gamma(1-\gamma)}{\gamma^2 N} = \frac{1-\gamma}{\gamma N} \tag{6}$$

which diverges to infinity as $\mathscr{O}(1/\gamma)$ when N is fixed and $\gamma \to 0$. An effective rare event simulation technique developed to decrease this computational burden is called splitting; refer to, e.g., Rubino and Tuffin (2009). Key to any splitting technique is the definition of an importance function $h : [0,T]\times\mathbb{R}^n \mapsto \mathbb{R}$ that assigns a value to each chain state (t,x) such that $h(t,x) \geq 1$ exactly when (t,x) corresponds to a rare event occurrence and $h(0,x_0) = 0$. Higher values of a suitable importance function correspond to a chain state from which the rare event is more likely. More precisely, the interval $[0,1]$ is divided into m subintervals with intermediate thresholds $0 = l_0 < l_1 < \cdots < l_m = 1$ where we define $T_k := \inf\{t > 0 : h(t,X^\varepsilon(t)) \geq l_k\}$ to be the first time of hitting the k-th level and $E_k := \{T_k < T\}$ to be the event that the k-th level is hit during $[0,T]$. Clearly, the value of interest $\mathbb{P}(E_m)$ is equal to γ.

Since $E_m \subset E_{m-1} \subset \cdots \subset E_0$, with $\mathbb{P}(E_0) = 1$, we can express γ as a product of m conditional probabilities $p_k := \mathbb{P}(E_k|E_{k-1})$, namely

$$\gamma = \prod_{k=1}^{m} \mathbb{P}(E_k|E_{k-1}), \tag{7}$$

which we will estimate separately. Independent sample paths from the conditional distribution of the entrance state $(T_{k-1}, X^\varepsilon(T_{k-1}))$, given E_{k-1}, would provide an estimate for p_k. However, we do not know this distribution for levels $k > 1$, and therefore we use its empirical distribution which is obtained from samples of the previous level. Proceeding recursively in this manner, where at each level k we estimate p_k by the proportion $\hat{p}_k$ of sample paths for which E_k occurs, we then have

$$\hat{\gamma} := \prod_{k=1}^{m} \hat{p}_k \tag{8}$$

as an unbiased estimator of γ for several variants of the splitting technique. One such technique is called Fixed Number of Successes (FNS) that repeatedly generates sample paths at each level k until a prespecified number r_k hits of the next level are observed. Then the conditional probabilities p_k are estimated by $\hat{p}_k := (r_k - 1)/(n_k - 1)$, with n_k the number of samples generated at level k. Amrein and Künsch (2011) show for this definition of $\hat{p}_k$ that (8) is indeed an unbiased estimator for γ and that, under ideal circumstances, the squared relative error of the FNS estimator diverges as $\mathscr{O}((\log \gamma)^2)$ when $\gamma \to 0$. This squared logarithmic divergence rate is slower than the divergence rate of the CMC squared relative error in (6), and we therefore use the FNS splitting technique to realize this potential gain of splitting.

Our choice for the importance function is crucial for variance reduction of the splitting estimator, which has been an active area of research. The goal is to select an importance function that splits sample paths which are more likely to hit the rare event set, where the levels should be chosen consistent with the most likely path to the rare event. Following Wadman, Crommelin, and Zwart (2016), we use a result from large deviations theory to find an asymptotic probability of the rare event in the limit of the rarity parameter ε. In particular, we are interested in the decay rate $I^*(s, x_s)$ of the limiting probability that, conditional on $X^\varepsilon(s) = x_s$ for general $s \in (0,T]$ instead of on $X^\varepsilon(0) = x_0$, the rare event $p(X^\varepsilon(\tau)) \geq P_{\max}$ will occur at some time $\tau \in (s,T]$ in the remaining time domain. The decay rate $I^*(s,x_s)$ is used to compute an approximate probability to hit the rare event given a realized chain state, which then serves as a proxy of the importance function for deciding whether or not to split the sample path at the corresponding time step.

From the derivations and results of Wadman, Crommelin, and Zwart (2016), we have that the most likely path from $X^\varepsilon(s) = x_s$ to the rare event is of the form

$$x(t) = (Ve^{Dt} - e^{-D(t-s)}Ve^{Ds})c + e^{-D(t-s)}(x_s - \mu) + \mu \tag{9}$$

for $t \in [s,T]$ and that the decay rate for general $s \in (0,T]$ is given by (5) where

$$g_{s,x_s}(\tau) = \frac{1}{2} \inf_{c \in \mathbb{R}^n : p(x(\tau)) = P_{\max}} c^\top e^{D\tau} \operatorname{Cov}(X^1(\tau - s)) e^{D\tau} c. \tag{10}$$

Assuming linear power flow equations $p(x) = v^\top x$, the latter minimization has the closed-form solution

$$g_{s,x_s}(\tau) = \frac{1}{2} \frac{\left(P_{\max} - v^\top \mu - v^\top e^{-D(\tau - s)}(x_s - \mu)\right)^2}{v^\top \operatorname{Cov} X^1(\tau - s) v}. \tag{11}$$

Next, to compute the rare event probability using splitting, we proceed by expressing γ as the product of conditional probabilities in (7) and by defining these conditional probabilities in terms of the decay rate in (5). This renders

$$\begin{aligned}\gamma = \mathbb{P}\left\{\exists s \in (0,T] : I^*(s, X^\varepsilon(s)) = 0 \;\middle|\; \exists s \in (0,T] : I^*(s, X^\varepsilon(s)) < \alpha I^*(0, x_0)\right\} \\ \times \mathbb{P}\left\{\exists s \in (0,T] : I^*(s, X^\varepsilon(s)) < \alpha I^*(0, x_0)\right\}\end{aligned}$$

for any threshold $\alpha \in (0,1)$, which obviously holds since the condition $\{\exists s \in (0,T] : I^*(s, X^\varepsilon(s)) < \alpha I^*(0,x_0)\}$ is a subset of the rare event $\{\exists s \in (0,T] : I^*(s,X^\varepsilon(s)) = 0\}$. Upon iterating this decomposition by defining thresholds $0 =: l_0 < l_1 < \cdots < l_m := 1$ and events

$$E_k := \left\{\exists s \in (0,T] : 1 - \frac{I^*(s, X^\varepsilon(s))}{I^*(0,x_0)} \geq l_k\right\},$$

for $k = 0, \ldots, m$, together with the facts $\mathbb{P}\{E_0\} = 1$, $\mathbb{P}\{E_m\} = \gamma$ and $E_0 \supset \ldots \supset E_m$, we conclude that the decomposition (7) holds. In turn, this decomposition suggests the large deviations based importance function $h : [0,T] \times \mathbb{R}^n \to \mathbb{R}$ defined by

$$h_{\mathrm{LD}}(t,x) := 1 - \frac{I^*(t,x)}{I^*(0,x_0)}. \tag{12}$$

This importance function is similar to that of Miretskiy, Scheinhardt, and Mandjes (2012), who choose an importance function equal to the exponential decay rate to estimate a probability of first entrance into a rare set and then prove asymptotic efficiency of their proposed Fixed Splitting scheme.

A direct implementation of the above decomposition approach can provide significant computational improvements through the benefits of splitting. As a specific example, described in Section 5.1 of (Wadman, Crommelin, and Zwart 2016), one can assume that the minimizer $\tau^* \in U := \{s, s+\Delta, \ldots, T\}$ lies on a uniform grid and then approximate the true optimum by computing the objective function $g_{s,x_s}(\tau)$ in (11) for each such grid point. The resulting approximation $\min_{\tau\in\{s,s+\Delta,\ldots,T\}} g(\tau)$ approaches the true optimum as Δ tends to 0. We shall refer to this as the Uniform Trials algorithm (UTA).

4 OPTIMIZING THE HITTING TIME OF THE RARE EVENT SET

Although significant improvements in efficiency are realized through splitting, UTA can still be computationally intensive because the optimization problem in (5) of $g_{s,x_s}(\tau)$ over $\tau \in (s,T]$ that defines the good-rate function $I^*(s,x_s)$ for any state $X(s) = x_s$ is in general non-convex. Thus, the UTA approach of direct-enumeration of g_{s,x_s} over a discretization of the range $(s,T]$ may require numerous evaluations of $g_{s,x_s}(\tau)$, potentially even outweighing the benefits of splitting. As an example, such evaluations of $g_{s,x_s}(\tau)$ cost on average 85% of the total CPU time in the experiments performed in Section 6.1. Our goal in this paper is to develop new algorithms and derive results that significantly improve upon the computational complexity of UTA by reducing the number of evaluations of $g_{s,x_s}(\tau)$.

Wadman, Crommelin, and Zwart (2016) show that the $\tau-$dependent matrix in the denominator in (11) is $\mathrm{Cov}(X^1(\tau - s)) = V - e^{-D(\tau-s)}Ve^{-D(\tau-s)}$, where V and D are symmetric matrices defined by the OU process data. Additionally, D is a positive diagonal matrix by assumption, and hence the matrix $e^{-D(\tau-s)}$ is a diagonal matrix consisting of terms of form $e^{-d_i(\tau-s)}$. Since the variable τ appears in (11) only via these terms, let $z = v^\top e^{-D(\tau-s)}$ and rewrite $g_{s,x_s} = (\bar{P} - z^\top \bar{x}_s)^2/(K - z^\top Vz)$, where $\bar{P} = (P_{\max} - v^\top \mu)$, $\bar{x}_s = (x_s - \mu)$ and $K = v^\top Vv$. Minimizing g_{s,x_s} over z is a program in a convex-fractional form (with a convex quadratic numerator and denominator) and hence can be reformulated as a convex program. However, since the power-injection vector v will have both positive and negative components, z is in general not a monotone function of τ, and so this technique does not lead to reducing the non-linear problem of minimizing g_{s,x_s} over τ to minimizing a convex program.

Differentiating with g_{s,x_s} respect to τ (using the symbol $\dot{x}$ to represent $\partial x(\tau)/\partial\tau$ for any $x(\tau)$),

$$\dot{g}_{s,x_s} = \frac{2(\bar{P} - z^\top \bar{x}_s)\left[(\bar{P} - z^\top \bar{x}_s)\dot{z}^\top Vz - \dot{z}^\top \bar{x}_s(K - z^\top Vz)\right]}{(K - z^\top Vz)^2}. \tag{13}$$

We seek a τ such that $\dot{g}_{s,x_s} = 0$ in order to find (first-order) optimal solutions to the optimization problem (5). By the definition of K and z, the denominator in (13) is always positive, and hence it is sufficient to seek a zero of the numerator. Denoting the numerator in (13) as $N(\tau)$, substitute $\dot{z} = -v^\top De^{-D(\tau-s)}$ and rearrange to get a form

$$\begin{aligned} N(\tau) \quad &= \quad \sum_i a_i e^{-b_i(\tau-s)} \quad + \quad \sum_j c_j e^{-l_j(\tau-s)}, \\ &= \qquad f(\tau) \qquad + \qquad h(\tau) \end{aligned} \tag{14}$$

where $b_i, l_j > 0$, and the coefficients $a_i > 0$ while $c_j < 0$.

The Tree-based Bisection Algorithm presented in Section 5 exploits this structure in determining the τ where $N(\tau) = 0$. Specifically, note that the function $f(\tau)$ is decreasing in τ while the function $h(\tau)$ is increasing in τ, and for any $\tau \in [\underline{\tau}, \overline{\tau}]$,

$$f(\overline{\tau}) + h(\underline{\tau}) \qquad \leq \qquad N(\tau) \qquad \leq \qquad f(\underline{\tau}) + h(\overline{\tau}). \tag{15}$$

Thus, if the upper bound $f(\underline{\tau})+h(\overline{\tau})<0$, then g_{s,x_s} is minimized over $[\underline{\tau},\overline{\tau}]$ at $\overline{\tau}$. Similarly, if the lower bound $f(\overline{\tau})+h(\underline{\tau})>0$, then g_{s,x_s} is minimized over $[\underline{\tau},\overline{\tau}]$ at $\underline{\tau}$. TBA calls this elimination rule as E, and checks this rule over subintervals of $(s,T]$ iteratively. If the function N does not satisfy E over an interval, it can be applied over a subinterval partition. TBA applies rule E to eliminate subintervals of $(s,T]$ till sufficiently small intervals have been identified as containing candidate optimal solutions. TBA then enumerates over a discretization of the remaining intervals in order to identify the optimal solution defining $I^*(s,x_s)$.

5 THE TREE-BASED BISECTION ALGORITHM

At each time step s of each sample path $X^{\varepsilon}(s)$ of the splitting simulation, the Tree-based Bisection algorithm (TBA) reduces the search space $(s,T]$ of the optimization problem in (5) of $g_{s,x_s}(\tau)$. To be more precise, the search space is reduced as follows:

Algorithm 1 We define the full search space $S:=(s,T]$ as the first interval to observe. The left and right endpoints of the interval are $\underline{\tau}:=s$, $\overline{\tau}:=T$, respectively. Then we perform the following steps:

1. We check whether the event $E:=\{f(\underline{\tau})+h(\overline{\tau})<0\}$ holds.
   (a) If E holds, then certainly $f(\tau)+h(\tau)<0$, so $\dot{g}_{s,x_s}(\tau)<0$ for all $\tau\in[\underline{\tau},\overline{\tau}]$ and thus $\arg\min g_{s,x_s}(\tau)=\overline{\tau}$. Therefore, we can remove this interval except the right endpoint from the search space by setting $S:=S\setminus[\underline{\tau},\overline{\tau})$.
   (b) If E does not hold, we check E for both the left and right half of the interval. That is, we repeat Step 1a after updating
      i. $\underline{\tau}:=\underline{\tau}$ and $\overline{\tau}=(\underline{\tau}+\overline{\tau})/2$, and
      ii. $\underline{\tau}=(\underline{\tau}+\overline{\tau})/2$ and $\overline{\tau}:=\overline{\tau}$.

In this way we proceed recursively, checking bound E for the left and right half of an interval whenever this bound was not valid for the entire interval. We kill a branch in this tree of checks whenever the interval width $\overline{\tau}-\underline{\tau}\leq C$ has become smaller than a prespecified minimum interval width C.

The resulting search space S is a union of intervals, which is a subset of the original search space $(s,T]$. If we assume just as in the UTA algorithm that $\arg\min g_{s,x_s}(\tau)$ is one of the $U=\{s,s+\Delta,\dots,T\}$, the minimization immediately reduces to computing $g(\tau)$ for all $\tau\in U\cap S$. Since $U\cap S$ may contain a much smaller number of points than the original U, TBA can be seen as an improvement to UTA, preserving accuracy and potentially saving a large amount of workload.

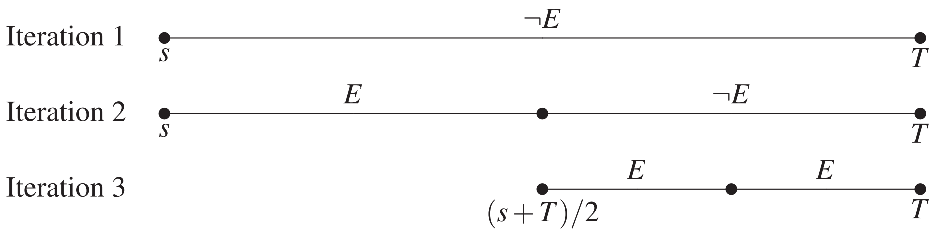

Figure 1: An example run of TBA, where after three iterations the search space $(s,T]$ is reduced to the single point T.

Note that Algorithm 1 ignores the opposite check whether $\{f(\overline{\tau})+h(\underline{\tau})>0\}$ as described in previous section. The reason is that simulation showed that the minimizer is often close to the end point T, and that proving $\dot{g}_{s,x_s}(\tau)<0$ on large intervals before the minimizer constitutes the lion's share of the computational gain.

6 EXPERIMENTS

We adopt the setting experiments on a IEEE 14 Bus Test Case in Section 6.1 of Wadman, Crommelin, and Zwart (2016) to compare computational efficiency. To keep this paper self-contained we briefly introduce all parameter values of the model and refer to Section 6.0 and 6.1 in Wadman, Crommelin, and Zwart (2016) for details.

The IEEE 14 grid has 20 connections connecting 14 nodes, of which 11 have nonzero power injections (Christie 2006). Although these power injections are constant in the original IEEE 14 Bus Test Case, we model nodes 2 and 3 as a two-dimensional OU process as in (1), so $n = 2$. This OU process fluctuates around the original deterministic power injections P^{det} by setting $\mu := x_0 := P^{\text{det}}$, $\varepsilon := 0.1$ and $\theta_i = 1 + (i-1)/(n-1)$. Matrix L is chosen as the lower Cholesky factor of covariance matrix $LL^\top := \text{diag}(\sigma)(\rho\mathbf{1}\mathbf{1}^\top + (1-\rho)I)\text{diag}(\sigma)$. Here correlation $\rho := 0.5$ reflects the typically positive correlation of power injections, $\mathbf{1} \in \mathbb{R}^n$ is a vector of ones and the vector σ of standard deviations is such that volatilities $\varepsilon\sigma_i := \varepsilon(1 + (i-1)/(n-1))$ of the marginal OU processes increase from 0.1 to 0.2.

We choose end time $T = 1$ hour and step size $\Delta = 0.01$ hour for the splitting simulation, so the experiment can be interpreted as an operational assessment of grid relibiality of the coming hour. For each connection $i \to j$ between grid nodes i and j, we estimate the probability γ of an overload in that connection, where the arrow distinguishes the direction of the overload:

$$\gamma = \begin{cases} \mathbb{P}\{\sup_{\tau\in(0,T]} v^\top X^\varepsilon(\tau) \geq P_{\max}\} & \text{if } i < j, \\ \mathbb{P}\{\inf_{\tau\in(0,T]} v^\top X^\varepsilon(\tau) \leq -P_{\max}\} & \text{if } i > j. \end{cases} \tag{16}$$

All experiments are performed on an Intel Core i7-4700MQ 2.40GHz computer in MATLAB R2015b.

We choose the maximum allowed power flow $P_{\max} = 1.5|v^\top\mu|$ equal to 1.5 times the average absolute power flow. Using $r = 100$ hits per level we estimate the overload probabilities of all connections for which the decay rate proxy $I^*(0, x_0)$ is larger than 10^{-20}, assuming the overload probabilities of the other connections are negligible.

6.1 Search Space Reduction

First we compare TBA with UTA by comparing the number of evaluations of g required to estimate each overload probability. Each row in Table 1 corresponds to (the overload probability of) another grid connection. The first column lists which overload probability as defined in (16) is concerned. Since no evaluation of g is required at the last splitting level, we ommitted overload probabilities of connections for which the splitting estimation would use only one level. The second level contains the number of levels m. For each connection we compute the average $\hat{\gamma}$ of 100 splitting estimates and its relative error $\text{SE}(\hat{\gamma})/\hat{\gamma}$, as listed in columns three and four, respectively. Note that TBA and UTA provide the exact same estimates and therefore they have equivalent accuracy. Furthermore, even when γ decreases by orders of magnitude, the resulting relative errors only increase modestly; in fact, one can easily verify that the divergence is consistent with the rate $\mathscr{O}((\log\gamma)^2)$ discussed in Section 3.

The fifth column displays the number n_{UTA} of evaluations of g required to obtain 1 splitting estimate, averaged over 100 splitting simulations, all using UTA. The sixth column similarly displays the number n_{TBA} of evaluations, but now using TBA instead of UTA. Therefore, the difference $n_{\text{UTA}} - n_{\text{TBA}}$ is exactly the number of evaluations of g *avoided* by TBA. We kill a branch of the tree in TBA whenever the interval width $\overline{\tau} - \underline{\tau} < C := \Delta$ is smaller than the grid step size, since removing smaller intervals will not necessarily remove a point from the set $\{s, s+\Delta, \ldots, T\}$ we have to compute g at. The last column reflects how much less evaluations of g TBA requires, by showing 1 minus the ratio between the counts of TBA and UTA. This column shows that TBA avoids evaluating g at around 90% of the grid points in this example. This decrease seems robust in the rarity of the event of the overload: even though $\hat{\gamma}$ varies from roughly one percent to almost 10^{-20}, this decrease stays between 89% and 92%.

Figure 2 shows for four connections which levels and which time steps contribute most to this decrease. More precisely, each bar shows the number of evaluations of g avoided by TBA at that time step s. Each bar

Table 1: Comparison of UTA and TBA in terms of the average number of evaluations required to find 1 splitting estimate.

	m	$\hat{\gamma}$	$\mathrm{SE}(\hat{\gamma})/\hat{\gamma}$	n_{UTA}	n_{TBA}	$1-\frac{n_{\mathrm{TBA}}}{n_{\mathrm{UTA}}}$
$5 \to 4$	3	1.2E−2	0.15	3.0E6	3.1E5	0.90
$1 \to 5$	4	1.4E−3	0.17	4.3E6	4.1E5	0.91
$3 \to 4$	5	2.5E−4	0.23	4.8E6	5.1E5	0.89
$11 \to 10$	6	6.3E−5	0.22	5.2E6	5.6E5	0.89
$2 \to 4$	15	5.5E−11	0.50	8.0E6	8.5E5	0.89
$9 \to 10$	19	3.2E−14	0.51	9.9E6	8.5E5	0.91
$6 \to 11$	26	5.2E−19	0.75	1.2E7	9.7E5	0.92

consists of *stacked* bars to further distinguish over levels, and all bar values in one figure sum exactly up to $n_{\mathrm{UTA}} - n_{\mathrm{TBA}}$ for the corresponding connection, as displayed in Table 1. First, one can see that most of the decrease is clearly achieved at lower levels. This can be attributed to the process X^{ε} starting at the mean $X^{\varepsilon}(0) = \mu$: at lower levels, $X^{\varepsilon}(s)$ will still be relatively close to the mean μ, implying that the most likely time to hit the rare event set will be close to end time T (see also Section 4.2 and Figure 4 in (Wadman, Crommelin, and Zwart 2016)). Simulation results confirm that at lower levels TBA typically removes the full interval (s,T) from the search space, leaving end time T only, so then apparently the righthand bound in (15) is sufficiently strict to solve optimization problem in (5) of $g_{s,x_s}(\tau)$ over $\tau \in (s,T]$ directly. Second, at the first level TBA clearly avoids less evaluations of g at later time steps. This is intuitive since at later time steps there simply are less evaluations to avoid by TBA: at every time step s UTA will evaluate g at $(T-s)/\Delta + 1$ hitting times (i.e., $\tau = s, s+\Delta, \ldots, T$). The approximate linearity of the decline of the first level bars suggests that TBA removes a relatively constant precentage of evaluations. At higher levels an approximate linear decline can only be distinguished at the last couple of time steps: at the start of a sample path, TBA starts avoiding evaluations after some time steps only. This can be attributed to sample paths entering higher levels at later time steps only, since it typically takes some time steps to enter a subsequent level.

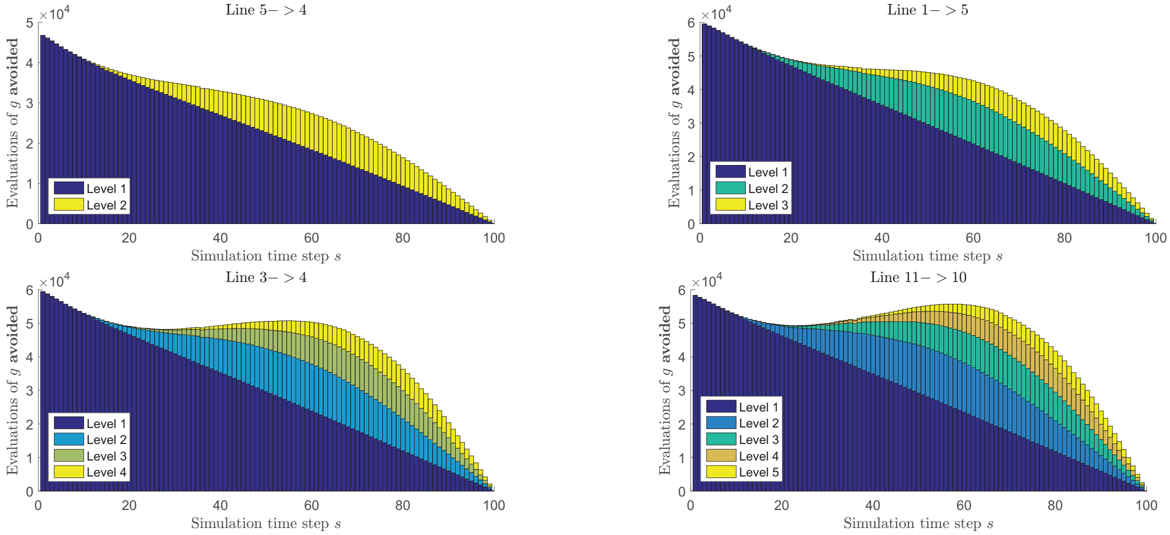

Figure 2: The number of evaluations of g avoided by TBA for each level (stacked) and at each time step s. Each figure corresponds to another connection.

In Figure 3 again the number of avoided evaluations are plotted, but now for each rare event hitting time τ instead of for each simulation time step s. More precisely, each bar shows the number of avoided evaluations of g for which the rare event hitting time is assumed to be τ. Again each bar consists of stacked bars to further distinguish over levels, and again all bar values in one figure sum up to $n_{\text{UTA}} - n_{\text{TBA}}$. To better understand these results, note that each plot contains exactly 99 stacked bars, since at the last time step $\tau = 100$ the bar is zero. This is intuitive since TBA is not capable of removing the last time step $\tau = T$ from the search space, as explained in Step 1a of Algorithm 1 in Section 5 and as illustrated in Figure 1. The plots in Figure 3 show that more evaluations of g at later hitting times are avoided than those at earlier hitting times. An explanation is that UTA evaluates g more often for larger τ, so that there are simply more evaluations that TBA can potentially avoid.

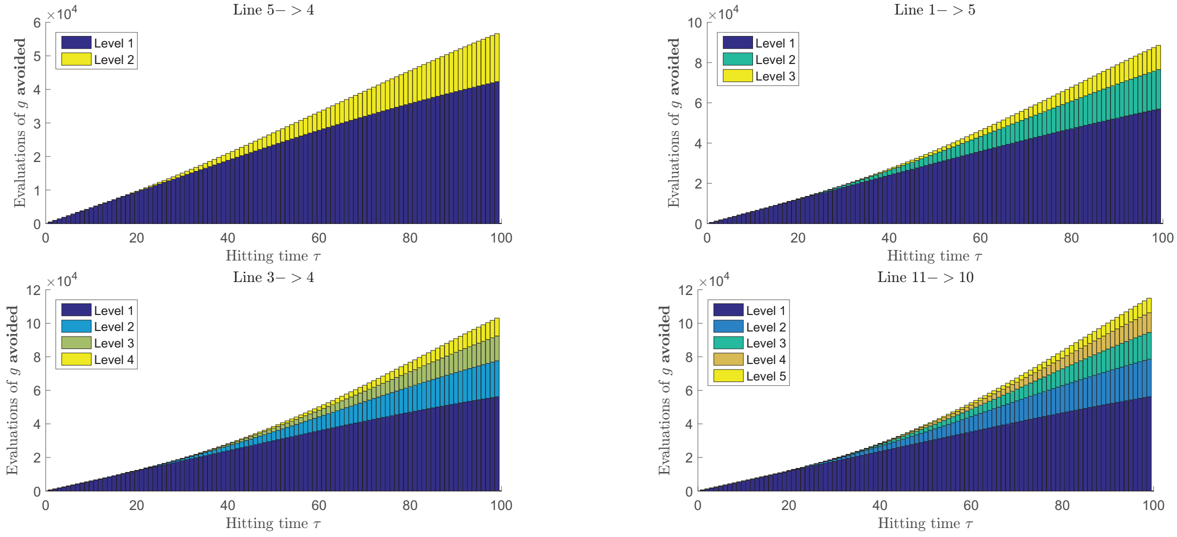

Figure 3: The number of evaluations of g avoided by TBA for each level (stacked) and at each rare event hitting time τ. Each figure corresponds to another connection.

6.2 Sensitivity to the Number of Time Steps

Suppose we would simulate grid reliability over a longer time horizon than $T = 1$ hour. Assuming a constant time step size Δ this will increase the number of time steps $M := T/\Delta$, and thus the number of evaluations of g that UTA will perform. We now investigate how evaluations TBA will require in comparison when increasing the number of time steps. We focus on the overload of connection $11 \rightarrow 10$ (see Table 1) only. To compare simulations with different numbers of time steps, ceteris paribus, we keep $T = 1$ constant but vary the time step size Δ. In this way the underlying continuous model and thus the rarity of the event of an overload, number of splitting levels, etcetera, remain unchanged. Table 2 shows that for $M \geq 50$ the ratio of evaluations avoided by TBA stays around 90% in this example. The suggested upper limit of this ratio can be attributed to the strictness of bound (15). Furthermore, it suggests robustness of the performance of TBA in the number of time steps.

6.3 Sensitivity to the Minimum Interval Size

We choose the time step size $\Delta = 0.01$ again and now investigate the influence of the minimum interval size C for TBA as described in Section 5. TBA will continue checking bound E on the left and right half of the current subinterval if its width is larger than this parameter C. Table 3 shows how much the ratio of evaluations avoided by using TBA increases in the example grid when choosing a smaller minimum interval width C. One can see from the table that choosing $C \leq 0.02$ smaller than twice the time step size does not substantially improve the search space reduction.

Table 2: The number of evaluations of g using either UTA and TBA, for different numbers of time steps M.

M	n_{UTA}	n_{TBA}	$1-\frac{n_{\text{TBA}}}{n_{\text{UTA}}}$
10	7.1E4	2.2E4	0.68
20	2.3E5	4.4E4	0.81
50	1.2E6	1.3E5	0.89
100	5.2E6	5.5E5	0.89
200	2.1E7	2.2E6	0.89
500	1.1E8	8.3E6	0.92
1000	4.6E8	4.5E7	0.90

Table 3: For different minimum interval sizes C, the fraction of evalutions avoided by using TBA instead of UTA during one splitting simulation. All displayed results times are averages over 10 simulation runs.

C	0.5	0.2	0.1	0.05	0.02	0.01	0.005	0.002	0.001
$1-n_{\text{TBA}}(C)/n_{\text{UTA}}$	0.48	0.64	0.76	0.83	0.88	0.89	0.89	0.90	0.90

6.4 Ongoing Research

Although a decrease of C will often further reduce the search space, the total workload does not necessarily decrease because evaluations of the bound $f(\underline{\tau})+h(\overline{\tau})$ may nullify the computational gains of avoided evaluations of g. We are exploring this interesting tradeoff as part of ongoing research. In addition, we are investigating the workload gains in larger grids with a higher number of uncertain power injections. Instead of focusing on one specific connection, the overload of any connection in due time will be of great importance for the grid operator. This generalization may involve repeating the computation of minimum good rate functions over all connections and then choosing the minimum over all connections, although we are exploring opportunities to further improve computational efficiency. In this setting there may be upfront calculations that avoid even more evaluations of g, for example by exploiting the fact that some lines are much less likely to overload than other lines.

7 CONCLUSION

We considered a power grid model with dependendent Ornstein-Uhlenbeck processes reflecting uncertain power generation and demand. When simulating overload probabilities with an asymptotically efficient splitting technique the computational bottleneck is the numerical optimization of the most likely time of the overload. We proposed to accelerate this optimization by first removing subintervals from the search space where the derivative of the objective function has a negative upper bound. We found in an experiment on an example power grid that around 90% of the numerical evaluations of the objective function are avoided in this way. Experimental results show that this reduction of a factor ten is robust in the rarity of the event of an overload, as well as in the number of simulation time steps.

REFERENCES

Allan, R. N., A. M. Leite da Silva, and R. C. Burchett. 1981. “Evaluation methods and accuracy in probabilistic load flow solutions”. *IEEE Transactions on Power Apparatus and Systems* 5 (PAS-100): 2539–2546.

Alzate, C., and M. Sinn. 2013. “Improved Electricity Load Forecasting via Kernel Spectral Clustering of Smart Meters”. In *2013 IEEE 13th International Conference on Data Mining*, 943–948.

Amrein, M., and H. R. Künsch. 2011. “A Variant of Importance Splitting for Rare Event Estimation: Fixed Number of Successes”. *ACM Transactions on Modeling and Computer Simulation* 21 (2): 13:1–13:20.

Christie, R.D. 2006. “Power Systems Test Case Archive”. Accessed Nov. 2016. http://www.ee.washington.edu/research/pstca/.

Esmaili, M., N. Amjady, and H. A. Shayanfar. 2010. “Stochastic Congestion Management in Power Markets Using Efficient Scenario Approaches”. *Energy Conversion and Management* 51:2285–2293.

Grainger, J. J., and W. D. Stevenson. 1994. *Power System Analysis*, Volume 621. McGraw-Hill.

Leite da Silva, A. M., and V. L. Arienti. 1990. “Probabilistic Load Flow by a Multilinear Simulation Algorithm”. In *IEE Proceedings C-Generation, Transmission and Distribution*, Volume 137, 276–282.

Low, S. H. 2014. “Convex Relaxation of Optimal Power Flow, Part I: Formulations and Equivalence”. *IEEE Transactions on Control of Network Systems* 1 (1): 15–27.

Miretskiy, D. I., W. R. Scheinhardt, and M. R. Mandjes. 2012. “On Efficiency of Multilevel Splitting”. *Communications in Statistics-Simulation and Computation* 41 (6): 890–904.

Rubino, G., and B. Tuffin. 2009. *Rare Event Simulation Using Monte Carlo Methods*. John Wiley & Sons.

Seifi, H., and M. S. Sepasian. 2011. *Electric Power System Planning: Issues, Algorithms and Solutions*. Springer Science & Business Media.

Su, C. L. 2005. “Probabilistic Load-Flow Computation Using Point Estimate Method”. 20 (4): 1843–1851.

Summers, T., J. Warrington, M. Morari, and J. Lygeros. 2015. “Stochastic Optimal Power Flow Based on Conditional Value at Risk and Distributional Robustness”. *Electrical Power & Energy Systems* 72:116–125.

Wadman, W. S., G. Bloemhof, D. T. Crommelin, and J. E. Frank. 2012. “Probabilistic Power Flow Simulation allowing Temporary Current Overloading”. In *Conference Proceedings on Probabilistic Methods Applied to Power Systems*, 494–499.

Wadman, W. S., D. T. Crommelin, and B. P. Zwart. 2016. “A Large-Deviation-Based Splitting Estimation of Power Flow Reliability”. *ACM Transactions on Modeling and Computer Simulation* 26 (4): 1–26.

Wood, A. J., and B. F. Wollenberg. 1996. *Power Generation Operation and Control*. New York: John Wiley & Sons, Inc.

Zhang, P., and S. Lee. 2004. “Probabilistic Load Flow Computation Using the Method of Combined Cumulants and Gram-Charlier Expansion”. *IEEE Transactions on Power Systems* 19 (1): 676–682.

AUTHOR BIOGRAPHIES

WANDER S. WADMAN is a postdoctoral researcher in the Smarter Energy and Environmental Sciences Department at the IBM T.J. Watson Research Center. His research interests are stochastic modeling, dynamical systems, Monte Carlo simulation and rare event simulation, with applications in power grid reliability and finance. He received his PhD in mathematics in 2015 at the University of Amsterdam, the Netherlands, while working at CWI Amsterdam in the Scientific Computing group. His email is wswadman@us.ibm.com and his web page is at sites.google.com/site/wanderswadman.

MARK S. SQUILLANTE is a Distinguished Research Staff Member and the Area Head of Stochastic Processes, Optimization and Control within the Mathematical Sciences Department at the IBM T.J. Watson Research Center, also serving as Director of the Center for Optimization under Uncertainty Research across IBM Research. His research interests broadly concern mathematical foundations of the analysis, modeling and optimization of complex stochastic systems. He is a Fellow of ACM and IEEE. His email address is mss@us.ibm.com and his web page is http://researcher.ibm.com/researcher/view.php?person=us-mss.

SOUMYADIP GHOSH is a Research Staff Member in the Mathematical Sciences Department at the IBM T.J. Watson Research Center. His current research interests lie in simulation based optimization techniques for stochastic optimization problems, with a focus on applications in Energy and Power systems and supply chain management. His email is ghoshs@us.ibm.com and his web page is at https://researcher.ibm.com/researcher/view.php?person=us-ghoshs.