

# A computational method for optimizing storage placement to maximize power network reliability

Debarati Bhaumik <sup>\*1</sup>, Daan Crommelin<sup>2</sup>, and Bert Zwart<sup>3</sup>

<sup>1</sup>CWI, Amsterdam

<sup>2</sup>CWI, Amsterdam and University of Amsterdam

<sup>3</sup>CWI, Amsterdam and Technical University Eindhoven

December 22, 2015

## Abstract

The intermittent nature of the renewable energy sources challenges the power network reliability. However these challenges can be alleviated by incorporating energy storage devices in the network. The aim of this study is to develop a computational technique which can find the optimal storage placement in the network with stochastic generations and demands such that the power network can be made more reliable. We use the probability of a line current violation as the reliability index of the network and find the optimal storage position so that this probability is minimal. We use the *simulated annealing* algorithm to minimize this probability under the variation of storage locations and capacities in the network, keeping the total storage capacity constant. In order to estimate the small probabilities of line current violations we use the *splitting technique* of rare-event simulation. We construct an appropriate *importance function* for splitting which enhances the efficiency of the probability estimator compared to the conventional Crude Monte Carlo estimator. As an illustration, we apply our method to the IEEE-14 bus network.

## 1 Introduction

The strive for reducing carbon footprints and a carbon free future has rapidly increased the usage of renewable energies in power networks. Renewable energy sources like photo-voltaic (PV) arrays and wind turbines are unpredictable in nature, which lead to intermittent power generation. The integration of intermittent renewable energy sources into the electrical power network challenges the network reliability.

---

\*Corresponding author. Email: [bhaumik@cwi.nl](mailto:bhaumik@cwi.nl)

However, network reliability can be enhanced by incorporating energy storage devices (batteries) in the network. The energy storage device acts as buffer by storing excess energy generated and delivering power when there is energy deficiency. The peak-shaving benefit of batteries have been studied long back in 1981 by [1]. Barton et al. in [2] developed a probabilistic method to study the ability of energy storage to increase penetration of intermittent energy sources in power grids.

Recent studies have investigated storage placement in the power network under the framework of optimal power flow ([3], [4], [5], [6], [7], [8], [9] and [10]). Ghofrani *et al.* in [4] minimized the hourly social cost using a market-based probabilistic optimal power flow with energy storage integration and wind generation. Gayme and Topcu proposed a solution strategy to solve the optimal control problem to investigate the effects of different energy storage capacities on generation costs and peak-shaving in [5]. However, they have neglected uncertainties due to fluctuations in demand and intermittency in generation. To study the energy storage dispatch and placement problem in power network with wind generation, Sjödin *et al.* proposed a risk-mitigated optimal power flow framework in [6]. Bose *et al.* studied optimal placement of large-scale energy storage in power grids using semidefinite relaxation of AC optimal power flow in [8]. Oh proposed a method to model the storage devices under the framework of DC optimal power flow in [9].

This study focuses on the optimal storage placement in a power network for reliable operation of the network. We model the line currents in the network according to the DC power flow equations and consider the Probability of Line Current Violation (PLCV) as the *reliability index* of the network. PLCV calculates the probability that one of the lines in the network has been overloaded, i.e., one of the line currents has exceeded its allowed maximum. These line current violations lead to physical damage to the lines because of the eventual temperature overload.

Given the distribution of the stochastic processes of the uncertain power injections and the total storage installation size, we aim to find the optimal placement of the storage devices in the network such that the PLCV is minimal. In order to do so, we use the *Simulated Annealing* algorithm to minimize the PLCV in the configuration space of different storage sizes and locations. We resort to simulated annealing because the configuration space of different storage locations and capacities is very large and the quantity we wish to minimize (the PLCV) is not guaranteed to be convex. Akker *et al.* have briefly discussed using simulated annealing for optimal storage placement in power network to minimize generation cost in [11]. Our study focuses on minimizing a cost function which is a rare-event probability.

For reliable operation of power network, PLCV should be small. The conventional Crude Monte Carlo (CMC) method is robust but becomes very inefficient for estimating small probabilities. To increase the efficiency we use the *Splitting Technique* for rare-event simulations [16]. We use a variant of splitting called the Fixed Number of Successes (FNS) proposed by Amrein and Künsch in [18]. Wadman *et al.* in [12] used FNS to estimate electrical grid reliability. The

efficiency of splitting is highly dependent on the *Importance Function* used [19]. We develop an appropriate IF for our problem. We verify numerically that our IF works well. A theoretical analysis would be interesting but is beyond the scope of the paper. Note that this is non-trivial since the power injections in the networks are not diffusion processes due to the buffers (storage devices), and as such, our setting does not fit in the framework of Wadman *et al.* in [20].

To the best of our knowledge, the combination of rare-event simulation with simulated annealing has not been carried out before. We apply our method to the IEEE-14 bus test case network. The uncertain power injections are modeled as Ornstein-Uhlenbeck processes. Given the fixed total storage instalment capacity, we find the optimal storage locations and capacities at each nodes of the network such that the PLCV is minimal. We start from different initial configurations of the storage locations and capacities to check the convergence of simulated annealing to the final storage configuration.

In section 2 we discuss the DC power flow equations, the stochastic process used to model the net power generation and the storage model. Section 3 defines the problem. Section 4 introduces simulated annealing algorithm and its various aspects used in the problem. Section 5 provides details of the FNS splitting technique and the IF used for the problem. Section 6 presents the simulation results showing how SA algorithm along with FNS minimizes PLCV efficiently for the given IEEE-14 bus test case network. Finally we conclude in section 7.

## 2 System set up

Let the power network be defined by a graph  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ , where  $\mathcal{N} := \{1, 2, \dots, N\}$  is the set of nodes (named as buses in power network) and  $\mathcal{E}$  is the set of edges (also called lines). We solve the DC power flow equations for calculating the nodal voltage angles and the line currents.

### 2.1 DC Power flow equations

In this section we will discuss the DC power flow equations which can be seen as a linear approximation to the AC power flow equations [13].

**Admittance matrix** Let  $y_{i,j}$  be the admittance of the line  $(i, j) \in \mathcal{E}$ , and  $y_{i,i}$  is the admittance-to-ground at bus  $i \in \mathcal{N}$ . If  $(i, j) \notin \mathcal{E}$ , then  $y_{i,j} = 0$ . The admittance matrix  $Y \in \mathbb{C}^{n \times n}$  is given by

$$Y_{i,j} = |Y_{i,j}|e^{i\alpha_{i,j}} = \begin{cases} -y_{i,j} & \text{if } i \neq j, \\ \sum_k y_{i,k} & \text{if } i = j. \end{cases} \quad (1)$$

The row sums of  $Y$  are zero and it is a symmetric matrix.  $G$  the conductance matrix and  $B$  the susceptance matrix are the real and imaginary parts of the admittance matrix  $Y$ , respectively, such that  $Y_{i,j} = G_{i,j} + iB_{i,j}$ .

**Bus voltages** Let  $V = (V_i, i \in \mathcal{N})$  be the vector of bus voltages. In polar form it is given by

$$V_i = |V_i|e^{i\theta_i} \quad (2)$$

where  $|V_i|$  is the voltage magnitude and  $\theta_i$  is the voltage angle of bus  $i$ .

**Current injections** Let  $I = (I_i, i \in \mathcal{N})$  be the vector of bus current injections which is given by Ohm's law,

$$I = YV. \quad (3)$$

**AC power flow equations** Let  $S = P + iQ \in \mathbb{C}^n$  be the vector of bus power injections. The AC power flow equations are

$$S_i = V_i I_i^* \quad (i \in \mathcal{N}). \quad (4)$$

Using Ohm's law (3) the above equation (4) can be written as (for  $i \in \mathcal{N}$ )

$$S_i = V_i \sum_{k=1}^N Y_{i,k}^* V_k^*. \quad (5)$$

Substituting (1) and (2) in (5) we get

$$S_i = \sum_{k=1}^N |V_i| |Y_{i,k}| |V_k| (\cos(\theta_i - \theta_k - \alpha_{i,k}) + i \sin(\theta_i - \theta_k - \alpha_{i,k})). \quad (6)$$

Comparing the real and the imaginary part we obtain the *power-flow equations* in polar form (for  $i \in \mathcal{N}$ )

$$P_i = \sum_{k=1}^N |V_i| |V_k| (G_{i,k} \cos(\theta_i - \theta_k) + B_{i,k} \sin(\theta_i - \theta_k)) \quad (7)$$

and

$$Q_i = \sum_{k=1}^N |V_i| |V_k| (G_{i,k} \sin(\theta_i - \theta_k) - B_{i,k} \cos(\theta_i - \theta_k)). \quad (8)$$

**DC Approximations** Under DC approximation the following assumptions are made:

1.  $|V_i| = 1$  for all  $i \in \mathcal{N}$ .
2. The voltage phase angle differences across a transmission line are small, i.e.,  $\forall (i, j) \in \mathcal{E}$  we have  $\sin(\theta_i - \theta_j) \approx (\theta_i - \theta_j)$  and  $\cos(\theta_i - \theta_j) \approx 1$ .
3. The resistive components of each line are ignored, i.e.,  $Y_{i,j} = iB_{i,j}$ .

4. The real power flow across lines are significantly larger than the reactive power flow, i.e.,  $P_{i,j} \gg Q_{i,j} \quad \forall (i,j) \in \mathcal{E}$ . Hence the reactive power is ignored.

These assumptions reduces the AC power flow equations into

$$P_i = \sum_k B_{i,k}(\theta_i - \theta_k) \quad (i \in \mathcal{N}), \quad (9)$$

known as the DC power flow equations. For any line  $(i,j) \in \mathcal{E}$ , the line current  $I_{i,j}$  flowing from  $i \rightarrow j$  is given by the Ohm's Law

$$I_{i,j} = Y_{i,j}(V_i - V_j) = B_{i,j}(\theta_i - \theta_j). \quad (10)$$

For the slack bus 1, we set  $\theta_1 = 0$  and  $P_1 = -\sum_{j=2}^N P_j$ .

## 2.2 Power generation

We consider every non-slack node ( $j = 2, \dots, N$ ) as a single household which has stochastic generation  $G(t)$  and demand  $D(t)$  at time  $t$  and produces net power  $P(t) := G(t) - D(t)$ . We model the net power generation at the  $i$ -th non-slack node  $P_i(t)$  as discretized Ornstein-Uhlenbeck (O-U) processes which are in fact AR(1) processes,

$$\Delta P_i(t) = \beta_i(\mu_i - P_i(t))\Delta t + \sigma_i \Delta W_i(t) \text{ for } i = 1, \dots, N-1. \quad (11)$$

where  $\mu_i$  is the long term mean,  $\beta_i$  is the mean reverting term,  $\sigma_i$  is the volatility term and  $W_i(t)$  denotes the Weiner process of the  $i^{\text{th}}$  O-U process. The values of these terms are determined later in section 6. Modelling power injections as O-U processes have been implemented by Wadman *et al.* in [12] and [20].

## 2.3 Storage model

We consider the storage devices (batteries) to be co-located with the stochastic non-slack nodes and are charged/discharged locally by the net power produced at each node. Let  $B_i(t)$  be the level of energy stored in the battery at time  $t$  at the  $i$ -th non-slack node, and it has a maximum capacity  $B_i^{\max}$ . The batteries are updated according to

$$B_i(t + \Delta t) = B_i(t) + p_i^B(t)\Delta t \quad \forall t \in [0, T] \quad (12)$$

where  $p_i^B(t)$  is the power flowing in/out of  $i$ -th the battery,  $\Delta t$  is the length of the time step and  $T$  is the time horizon of interest. The batteries are bounded by their corresponding capacity constraints

$$0 \leq B_i(t) \leq B_i^{\max} \quad \forall t \in [0, T] \quad (13)$$

and total installation capacity constraint

$$\sum_i B_i^{\max} = B^{\text{tot}}. \quad (14)$$

Let  $P_i^B(t)$  be the power generated by the  $i$ -th battery and is given by  $P_i^B(t) = -p_i^B(t)$ . This is because,  $p_i^B(t) > 0$  implies the battery is getting charged and it is consuming power and  $p_i^B(t) < 0$  implies the battery is discharging and is generating power.

### 2.3.1 Switching of the battery

The battery charging/discharging depends on the power flowing in/out of the battery  $p_i^B(t)$  which is given by

$$p_i^B(t) = \begin{cases} P_i(t) & \text{if } 0 \leq X_i(t) \leq B_i^{\max} \\ (B_i^{\max} - B_i(t))/\Delta t & \text{if } X_i(t) > B_i^{\max} \\ -B_i(t)/\Delta t & \text{if } X_i(t) < 0. \end{cases} \quad (15)$$

where  $X_i(t) = P_i(t)\Delta t + B_i(t)$  for  $i = 1, \dots, N - 1$ . Hence the battery charging/discharging is dependent on the net power generated by the stochastic non-slack buses and the state of the battery. The above equation (15) ensures that (13) is true. To keep the storage model simple we neglect the *ramp constraints*, the imposed maximal charging/discharging rate on the storage device in this study.

## 3 Problem description

Our aim is to find the optimal battery locations and capacities at each node to ensure a reliable operation of the network. We consider the *Probability of Line Current Violation* (PLCV) as the reliability index of the network.

PLCV is defined as the probability that any one of the line currents violate its given line constraint maximum at any time  $t \in [0, T]$ , i.e,

$$\gamma := \mathcal{P}\{\exists(i, j) \in \mathcal{E} : \sup_{t \in [0, T]} |I_{i,j}(t)| \geq I_{i,j}^{\max}\}. \quad (16)$$

In the above equation  $I_{i,j}$  is the current flowing between nodes  $i$  and  $j$  and  $I_{i,j}^{\max} > 0$  is the maximum current carrying capacity the edge connecting nodes  $i$  and  $j$ .

To solve the optimal storage (battery) placement problem and calculating PLCV we use novel a combination of two algorithms namely the *simulated annealing* algorithm and the *splitting* technique for rare-event simulations, respectively, discussed in the subsequent sections.

## 4 Simulated annealing (SA)

We wish to find the optimal location and capacities of the battery in the network such that PLCV is minimal. We do not expect PLCV to be convex. Also, the configuration of space of battery locations and capacities to grow exponentially

with the number of nodes in the network. To overcome these problems we use the Simulated Annealing (SA) [14] algorithm to minimize PLCV.

SA algorithm is a metaheuristic algorithm designed for optimization to find the global minimum of a function. It generalizes the ideas developed from the Metropolis algorithm. The main aim of the algorithm is to perform a local search in the the solution space  $\mathbf{X}$  of the problem to find the global minimum of a desired *cost function*  $f(\mathbf{X})$ . Annealing is a physical process of heating a solid to very high temperature to randomize the particles of the solid, then it is cooled by slowly lowering the temperature of the solid. This slow cooling helps the particles to arrange themselves in a low energy state of a corresponding lattice. This helps in eliminating point defects in solids.

The SA algorithm described in Algorithm 1 is based on the following search principle :

- 1 Start with an initial solution and consider it as the best solution  $\mathbf{X}_{\text{best}}$ . Initialize  $T_c$ , the temperature of the *acceptance probability* of a *bad* solution .
- 2 Randomly select a new solution  $\mathbf{X}^*$  in the neighbourhood of the previously obtained best solution.
- 3 If the new solution is better than the previously found best solution, i.e, if  $\Delta E = f(\mathbf{X}^*) - f(\mathbf{X}_{\text{best}}) < 0$ , then consider the new solution as the best solution  $\mathbf{X}_{\text{best}} = \mathbf{X}^*$ .
- 4 If not, i.e,  $\Delta E = f(\mathbf{X}^*) - f(\mathbf{X}_{\text{best}}) > 0$ , then accept the new *bad* solution as the best solution with a probability  $\exp(-\Delta E/T_c)$ .
- 5 Slowly cool the temperature of the acceptance probability (decrease  $T_c$ ).
- 6 Repeat from 2 until the *stopping criterion* is reached.
- 7 Report the best solution.

The cooling law has to be chosen carefully to allow the algorithm sufficiently explore the region around the initial guess. If the cooling is too fast, the system will get stuck in the nearest local minima and the algorithm may not converge. If the cooling is too slow, the algorithm spends a lot of time in useless explorations which slows down the process. Usually, an exponential decrease is considered for the cooling by multiplying the current temperature with a constant, i.e.  $T_c^{\text{new}} = \kappa T_c^{\text{old}}$  where  $0 < \kappa < 1$ .

#### 4.1 Cost function: $\log(\gamma)$

In this study we want to minimize  $\gamma$  (PLCV (16)) in the battery locations and capacities configuration space. The values of  $\gamma$ 's are typically small and while minimizing can go down to  $\sim 10^{-5} - 10^{-7}$  or smaller depending on the total installation capacity of the battery. The *acceptance probability* of the bad solution not only depends on  $T_c$  but also on the difference of the function values

---

**Algorithm 1** Simulated Annealing algorithm

---

```
1:  $T = \text{initialize}$ 
2:  $\mathbf{X}_{\text{best}} = \text{initialize}$ 
3: while Stopping criterion do
4:    $\mathbf{X}^* = \text{Randomly move to new solution in the configuration space}$ 
5:    $\Delta E = f(\mathbf{X}^*) - f(\mathbf{X}_{\text{best}})$ 
6:   if  $\Delta E < 0$  then
7:      $\mathbf{X}_{\text{best}} = \mathbf{X}^*$ 
8:   else
9:      $r = \text{random uniform number} \in [0, 1]$ 
10:    if  $r < \exp(-\Delta E/T)$  then
11:       $\mathbf{X}_{\text{best}} = \mathbf{X}^*$ 
12:    end if
13:  end if
14:   $T = \kappa T$  for  $0 < \kappa < 1$ .
15: end while
16: Report best solution  $\mathbf{X}_{\text{best}}$ .
```

---

for the new solution and the previously found best solution  $\Delta E = \gamma(\mathbf{X}^*) - \gamma(\mathbf{X}_{\text{best}})$ . As the  $\gamma$ 's are very small, their differences are also small hence the acceptance probability becomes large and the algorithm accepts too many bad solutions and might never converge. So, instead of minimizing  $\gamma$  we minimize  $\log(\gamma)$  such that  $\Delta E = \log(\gamma(\mathbf{X}^*)) - \log(\gamma(\mathbf{X}_{\text{best}}))$  does not take very small values and the algorithm does not accept too many bad solutions.

## 4.2 Random moves in battery configuration space

We evaluate  $\gamma$  for different battery locations and capacities in the SA algorithm and then minimize  $\log(\gamma)$ . To move randomly in the solution configuration space, we randomly select two non-slack nodes  $(i, j) \forall i \in \mathcal{N}/\{1\}$  and  $\forall j (\neq i) \in \mathcal{N}/\{1\}$ . Then exchange  $m_B$  blocks of battery unit  $\Delta B$  between the two chosen nodes such that conditions (13) and (14) are satisfied. Initially we start the algorithm by exchanging  $m_B = N_B$  blocks of the battery unit and gradually reduce  $m_B$  till it is equal to 1. The gradual reduction of  $m_B$  depends on the decrease of  $\gamma$ . As the value of  $\gamma$  reduces by a factor of 10,  $m_B$  is decreased by  $\Delta m$ . This ensures that when the desired minimum is reached the system does not jump out of the minimum. The values of  $\Delta B$ ,  $N_B$  and  $\Delta m$  will be discussed in subsequent sections.

## 4.3 Stopping criterion for SA

We enforce three simultaneous *stopping criteria* for SA algorithm:

1. The number of iterations,  $n_{\text{iter}}$ , exceeds a pre-defined threshold value,  $n_{\text{max}}$ , i.e.,  $n_{\text{iter}} \geq n_{\text{max}}$ .



2. The difference between  $n_{\text{iter}}$  and the number of solutions the algorithm has accepted,  $n_a$ , exceeds a maximum value,  $n_d$ , i.e.,  $n_{\text{iter}} - n_a \geq n_d$ .
3. The improvements in  $\gamma$  have reached a desired minimum,  $\epsilon$ , i.e.,  $\gamma_{\min} = \max_{n_m} |\gamma^{n_a} - \gamma^{n_a - n_m}| \leq \epsilon$ , where  $\gamma^{n_a}$  is  $\gamma$  for the accepted iteration  $n_a$ .

If any of the stopping criteria is true the algorithm stops.

## 5 Fixed number of successes splitting technique

In our model we are interested in estimating PLCV and then minimize the logarithm of PLCV. It is expected that PLCV will take very small values during minimization. The Crude Monte Carlo (CMC) estimations of these small probabilities will become computationally very expensive and thus we use the *splitting technique* for rare-event simulations. In this section we give a brief overview of splitting technique, following [15] closely.

### 5.1 Splitting technique for rare-event simulation

Let  $(\Omega, \mathcal{F}, \mathcal{P})$  be a probability space and  $R \in \mathcal{F}$  be the rare event set of interest [16]. The probability of reaching the rare-event set  $R$ ,  $\bar{\gamma} = \mathcal{P}(R)$  will be small. The CMC estimator of  $\mathcal{P}(R)$  is given by

$$\tilde{\gamma} := \frac{1}{M} \sum_{j=1}^M \mathbb{1}_{\{\text{sample } j \text{ reaches } R\}}, \quad (17)$$

where  $M$  is the number of samples generated.

The *squared relative error* of the CMC estimator is given by

$$\text{SRE}(\tilde{\gamma}) := \frac{\text{Var}(\tilde{\gamma})}{\tilde{\gamma}^2} = \frac{\tilde{\gamma}(1 - \tilde{\gamma})/M}{\tilde{\gamma}^2} = \frac{1 - \tilde{\gamma}}{\tilde{\gamma}M}. \quad (18)$$

CMC probability estimator becomes unreliable for small  $\tilde{\gamma}$ . This is because, the  $\text{SRE}(\tilde{\gamma}) \rightarrow \infty$  as  $\tilde{\gamma} \rightarrow 0$  for fixed  $M$ . Otherwise to achieve an acceptable SRE we need very large values of  $M$ . For example, to estimate probabilities smaller than  $10^{-4}$  one needs  $M \gtrsim 10^6$  CMC samples for achieving  $\text{SRE} \approx 0.01$ .

Computational workload for estimation small probabilities of rare-events can be reduced by using rare-event simulations techniques like Importance Sampling and Splitting [16]. We use the *splitting technique* for estimating  $\tilde{\gamma}$  in our simulations [17]. In splitting, the distance to the rare-event set is measured in terms of the *Importance Function* (IF). The sample paths of the stochastic processes involved are split into multiple copies at various levels of the IF till the rare event set is reached. The probability  $\tilde{\gamma}$  is decomposed into the product of several conditional probabilities which are non-rare and are hence less computationally intensive to calculate. The most important ingredient of splitting is to find an appropriate IF for the problem.

Let  $\mathbf{X}$  be a vector of Markov processes with state space  $\xi$

$$\mathbf{X}(t) := (X_1(t), \dots, X_n(t)), \forall t \geq 0.$$

The importance function,  $\phi(\mathbf{X}(t)) : \xi \rightarrow \mathbb{R}$ , assigns importance values to  $\mathbf{X}(t)$ . Let  $R_{\phi, L, t}$  be the rare event set defined in terms of  $\phi$  as

$$R_{\phi, L, t} = \{\mathbf{X}(t) \in \xi : \phi(\mathbf{X}(t)) \geq L\}.$$

We are interested in the rare event

$$R = \{\exists t \leq T : R_{\phi, L, t} \text{ holds}\}. \quad (19)$$

For splitting, we divide the interval  $[0, L]$  into  $m$  sub-intervals with boundaries  $0 = l_0 < l_1 < \dots < l_m = L$ . Let the time of hitting the  $k$ -th level be  $T_k = \inf\{t > 0 : \phi(\mathbf{X}(t)) \geq l_k\}$  and the event that the  $k$ -th level is hit during  $[0, T]$  be  $H_k = \{T_k < T\}$ . Therefore,  $\mathcal{P}(H_m) = \bar{\gamma}$  and  $\mathcal{P}(H_0) = 1$ . As  $H_m \subset H_{m-1} \subset \dots \subset H_1 \subset H_0$ , we have

$$\bar{\gamma} = \mathcal{P}(R) = \prod_{k=1}^m \mathcal{P}(H_k | H_{k-1}) = \prod_{k=1}^m p_k,$$

where  $p_k := \mathcal{P}(H_k | H_{k-1}) = \mathcal{P}(H_k) / \mathcal{P}(H_{k-1})$ . Probability of hitting each of the  $k$ -th level,  $p_k$ , is estimated separately by generating independent sample paths from the distribution of the entrance state  $G_{k-1} := (T_{k-1}, \mathbf{X}(T_{k-1}))$  conditioned on  $H_{k-1}$  at the threshold level  $l_{k-1}$ . The empirical distribution  $\hat{G}_k$  is an estimate of the entrance distribution  $G_k$  which is obtained from  $H_k$ . Thus we can proceed recursively, replacing  $\hat{G}_{k-1}$  for  $G_{k-1}$  and estimate  $p_k$  at each level  $k$  by the proportion of level hits

$$\hat{p}_k = S_k / N_{k-1} \text{ for all } S_k > 0, \quad (20)$$

with  $S_k$  are the number of sample paths where  $H_k$  occurs and  $N_k$  is the total number of sample paths at level  $k$ .  $\hat{\gamma}$  is estimated by the product of  $\hat{p}_k$ 's :

$$\prod_{k=1}^m \hat{p}_k = \prod_{k=1}^m \frac{S_k}{N_{k-1}}. \quad (21)$$

We use a variant of splitting called the Fixed Number of Successes (FNS) proposed by Amrein and Kunsch [18]. For FNS the number of hits per level  $S_k$  is kept fixed. The process is independently repeated by selecting an entrance state at random and simulating the process from the selected state upto  $\min\{T_k, T\}$  until  $S_k$  hits are observed. Using FNS one can avoid path extinction or explosion, however, computational effort is compromised. The unbiased estimator of the rare-event probability is given by

$$\hat{\gamma} := \prod_{k=1}^m \tilde{p}_k = \frac{S_k - 1}{N_{k-1} - 1}. \quad (22)$$

The unbiased estimator for the variance  $\text{Var}(\hat{\gamma})$  is not known for the FNS method. However, under the assumption that the conditional hitting probability does not depend on the entrance states of the previous stage,

$$\begin{aligned} \mathcal{P}(H_k|H_{k-1}, (T_{k-1}, \mathbf{X}(T_{k-1}))) &= \mathcal{P}(H_k|H_{k-1}) \\ &(\forall(T_{k-1}, \mathbf{X}(T_{k-1})), \forall k), \end{aligned}$$

the squared relative error of  $\hat{\gamma}$  can be bounded :

$$\text{SRE}(\hat{\gamma}) \leq \prod_{k=1}^m \left( \frac{1}{S_k - 2} + 1 \right) - 1. \quad (23)$$

## 5.2 Importance function for calculating PLCV

The efficiency of splitting is significantly determined by the IF [19]. We take the maximum of the ratio of the absolute value of line currents and their respective maximum line current capacity as the importance function  $\phi(t)$ . This makes  $\phi(t)$  an increasing function in  $[0,1]$  and it is given by

$$\phi(I_{i,j}(t)) := \max_{(i,j) \in \mathcal{E}} \frac{|I_{i,j}(t)|}{I_{i,j}^{\max}}. \quad (24)$$

At time  $t$ , for any  $(i,j) \in \mathcal{E}$  if  $|I_{i,j}(t)| \geq I_{i,j}^{\max}$  implies  $\phi(I_{i,j}(t)) \geq 1$ , signifying that the rare event is hit, i.e., one of the line currents has exceeded it's line capacity.  $\phi \rightarrow 1$  corresponds to approaching the rare event set. A similar IF was used in [12].

## 6 Simulation results

In this section we apply the SA algorithm and FNS to find the optimal storage position in a power network by minimizing PLCV.

### 6.1 Simulation parameters

We first discuss the different parameters chosen for our simulations. For the O-U processes (11), we consider all the long term mean terms  $\mu_i$  to be zero, which implies on an average at each non-slack node the power demand is compensated by the local power generation. The mean reverting terms  $\theta_i = 1 + (i-1)/(N-2)$  for  $i = 1, \dots, N-1$  increases from 1 to 2 with  $i$ . The volatility terms  $\sigma_i$  are calculated from the long-term standard deviations of the O-U process  $\text{std}(P_i(t)) = \sigma_i/\sqrt{2\theta_i}$ . The values of  $\text{std}(P_i(t))$  will be discussed in the subsequent sections. Currently, we only consider uncorrelated O-U processes for our assessment.

We perform our simulations for  $T = 24$  hours and  $\Delta t = 0.01$  hours. The initial state of the batteries are taken as  $B_i(0) = B_i^{\max}/2 \forall i \in \mathcal{N}/\{1\}$ , i.e, they are half-filled.

For performing FNS we first calculate the number of levels  $m$  by the pilot run such that  $\bar{p}_k$  is nearly equal to the optimal value of  $p_{\text{opt}} \approx 0.2032$  [18]. For the pilot run we use  $R_k = 50$  for all  $k$ . For the final run we calculate  $R_k$  from (23) such that the  $\text{SRE}(\hat{\gamma}) \leq 0.03$ . In order to obtain an accurate estimate of the probabilities, FNS is repeated  $n = 30$  times (suggested by [21]) to calculate the mean of the estimator

$$\hat{\gamma}_n := \frac{1}{n} \sum_{i=1}^n \hat{\gamma}_i. \quad (25)$$

and the squared relative error of the mean computed from the  $n$  samples,

$$\text{SRE}(\hat{\gamma}_n) := \frac{1}{n} \text{SRE}(\hat{\gamma}_i). \quad (26)$$

For SA we take initial temperature  $T_c = 1$  and  $\kappa = 0.99$ . For the stopping criterion we take  $n^{\text{max}} = 1000, n_d = 300$  and  $\epsilon = 10^{-7}$ .

## 6.2 IEEE-14 bus test case

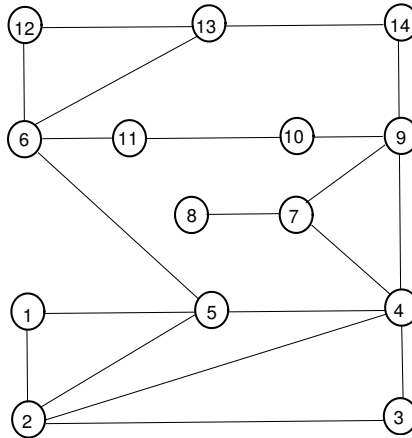


Figure 1: IEEE-14 bus line diagram showing the bus numbers and line connections. Bus 1 is the slack bus.

Fig. 1 shows the line diagram of the IEEE-14 bus test case. We test our algorithm for different scenarios for the IEEE-14 bus test case. The different scenarios being different maximum current carrying capacity of the lines  $I_{i,j}^{\text{max}} \forall (i,j) \in \mathcal{E}$  and different standard deviations  $\text{std}(P_i(t))$  of the O-U processes of the non-slack nodes and total installation capacity of storage  $B^{\text{tot}}$ . We use the MATPOWER package [22] of MATLAB for the topological details (admittance matrix) of the IEEE-14 bus test case network. However, the maximum line current carrying capacity is not set by the test case. We will discuss the values of the maximum line current carrying capacities in the subsequent sections.

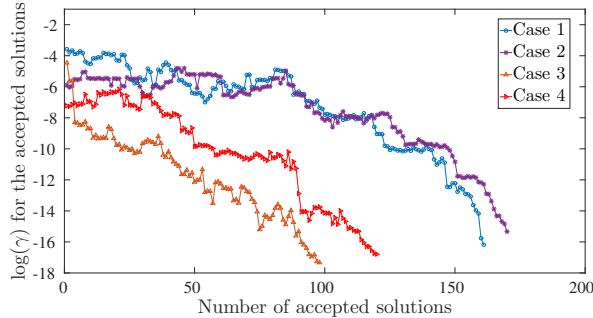


Figure 2:  $\log(\gamma)$  versus the number of accepted solutions for different starting configuration of battery locations and capacities with  $B^{\text{tot}} = 13000$  p.u. (Example 1).

### 6.2.1 Example 1

For this case the values of  $I_{i,j}^{\text{max}} \forall (i,j) \in \mathcal{E}$  were obtained by simulating a long time-series,  $T=10^4$  hours, for the system. The maximum value of line currents attained from the time-series run was taken to be the allowed maximum of each lines. For the  $\text{std}(P_i(t))$  we use the net power injections at each non-slack node from MATPOWER, the values range from 1 to 95 p.u. (where p.u. is per unit value of the quantity of interest).

We start from different random initial configurations (cases 1-4) of the battery locations and capacity to minimize  $\log(\gamma)$ . The total installation capacity of storage  $B^{\text{tot}} = 13000$  p.u. For the random movement of the algorithm in the battery configuration space (section 4.2) we take the battery unit  $\Delta B = 100$  p.u., the initial number of blocks exchanged  $N_B = 5$  and  $\Delta m = 1$ .

From Fig. 2 we observe that  $\gamma$  has reduced by roughly a factor of  $e^{10}$ . In Fig. 3 we compare the initial and final configurations of the battery position and capacities. It is observed that for all the cases in the final configurations about 35 percent of  $B^{\text{tot}}$  is placed at bus 3. One plausible reason for this final configuration can be the fact that  $\frac{\text{std}(P_3(t))}{\sum_{i=2}^{14} \text{std}(P_i(t))} = 0.365$ . We note that the final battery sizes at all the other nodes (apart from 3) are different for all the four cases and are not as consistent as node 3. Notwithstanding, for all four cases  $\gamma$  is reduced to very small values, see Fig. 2.

In Fig. 4 we compare the accepted configuration solutions with the total configuration solutions the algorithm has searched for (from case 3).

### 6.2.2 Example 2

In this example, we take all the non-slack nodes to be similar, i.e., all the OU processes have same standard deviations  $\text{std}(P_i(t)) = 10$  p.u.  $\forall i \in \mathcal{N}/\{1\}$ . Unlike Case 1 we randomize the  $I_{i,j}^{\text{max}}$ . To do this, we perform a long time series run,  $T=10^4$  hours, for the system to calculate the maximum current flow

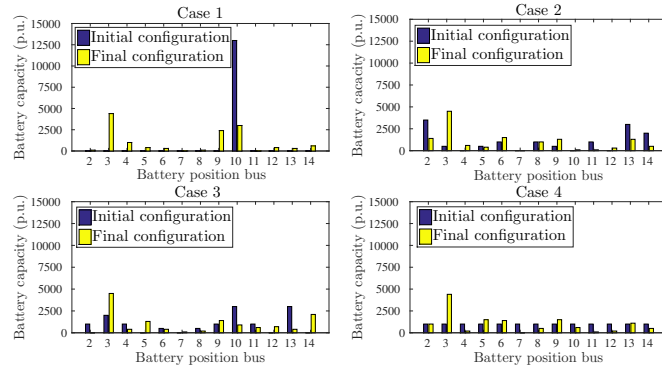


Figure 3: Comparing the initial and the final configuration of the battery locations and capacities for four different initial states from Fig. 2.

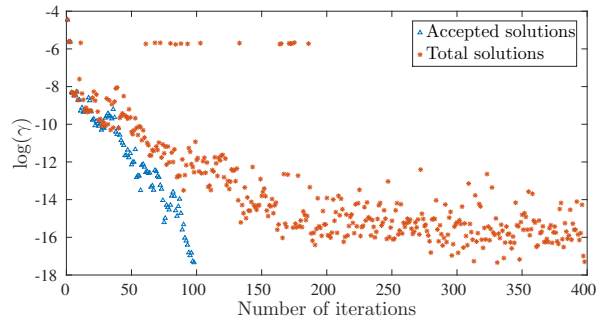


Figure 4:  $\log(\gamma)$  versus the number of accepted solutions and total number to solutions the SA algorithm searched for (Case 3 of Example 1).

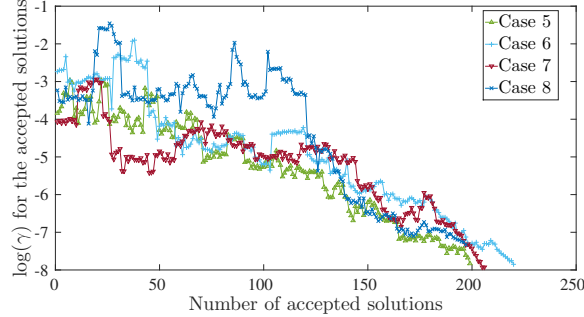


Figure 5:  $\log(\gamma)$  versus the number of accepted solutions for different initial configuration of battery locations and capacities with  $B^{\text{tot}} = 2600$  p.u. (Example 2).

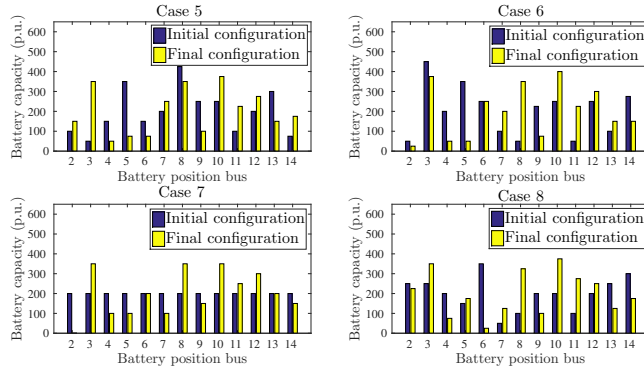


Figure 6: Comparing the initial and the final configuration of the battery locations and capacities for the four different cases in Fig. 5.

through each lines, and multiply them with uniform random numbers between  $[0.5, 1]$  to obtain  $I_{i,j}^{\text{max}}$ . We take the battery unit  $\Delta B = 12.5$  p.u., the initial number of blocks exchanged  $N_B = 8$  and  $\Delta m = m_B/2$  (section 4.2).

Fig. 5 shows the minimization of  $\log(\gamma)$  for four different initial configurations. We find that  $\gamma$  is reduced by a factor of  $e^4$ . By comparing the initial and final configurations of the battery locations in Fig. 6, we find that buses 3, 8 and 10 require each around 15 percent of  $B^{\text{tot}}$  for all the cases for this minimization.

### 6.2.3 Example 3

In the example we make all the non-slack nodes and connection lines equivalent to study the effect of number of connections at nodes on the optimization. To do so, we take  $\text{std}(P_i(t)) = 10$  p.u.  $\forall i \in \mathcal{N}/\{1\}$  and  $I_{i,j}^{\text{max}} = 50$  p.u.  $\forall (i, j) \in \mathcal{E}$ .

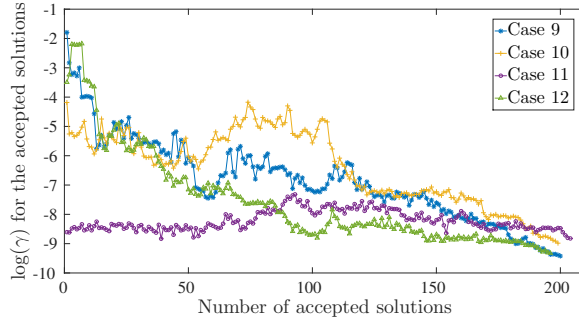


Figure 7:  $\log(\gamma)$  versus the number of accepted solutions for different starting configuration of battery locations and capacities with  $B^{\text{tot}} = 2600$  p.u. (Example 3).

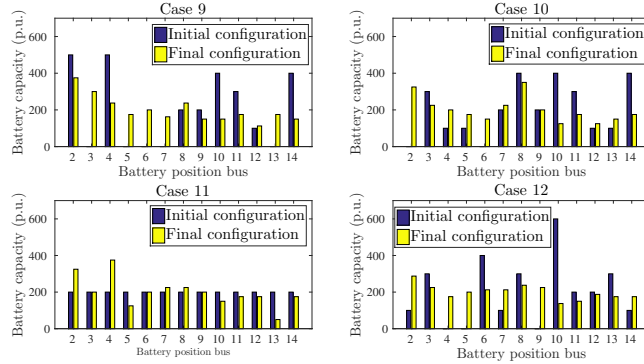


Figure 8: Comparing the initial and the final configuration of the battery locations and capacities for four different cases from Fig. 7.

We take the battery unit  $\Delta B = 12.5$  p.u., the initial number of blocks exchanged  $N_B = 8$  and  $\Delta m = m_B/2$  (section 4.2). Fig. 7 shows the optimization of  $\log(\gamma)$  for various cases, where each case represents a different starting configuration. The initial and final configurations of the battery placement is shown in Fig. 8. Notice for Case 11 when the batteries are placed equally at the non-slack buses the SA algorithm is not able to minimize  $\log(\gamma)$  further. This hints towards the fact that equally placing the batteries at the non-slack buses is near optimal solution to the problem. Equal battery placement being the near optimal solution shows that number of connections at nodes is not important for minimizing  $\log(\gamma)$ .

We now investigate how  $\log(\gamma)$  varies with  $B^{\text{tot}}$ . In order to do that we place the batteries equally at the non-slack buses. We do this because, Fig. 7 suggests that equally placing the batteries at the non-slack buses is near the



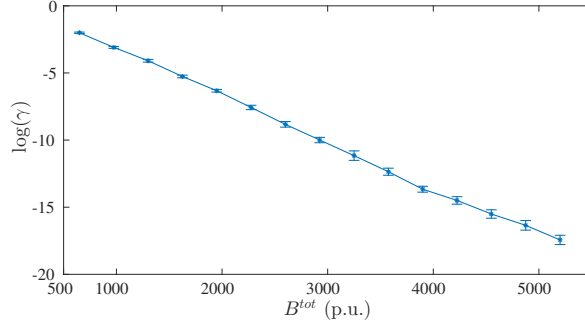


Figure 9:  $\log(\gamma)$  versus  $B^{\text{tot}}$ . The error bars represents 95 percent confidence interval of  $\log(\gamma)$ .

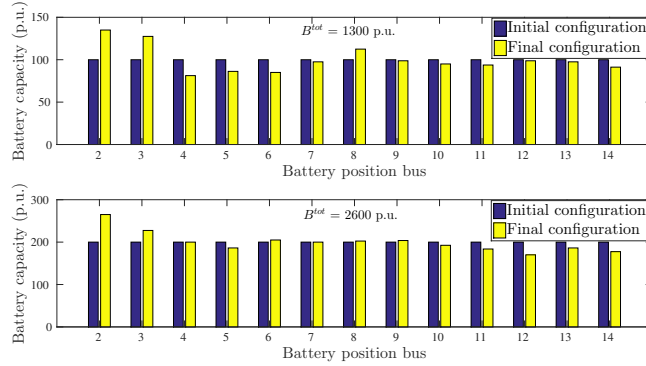


Figure 10: Comparing the initial and the final configuration of the battery locations and capacities for  $B^{\text{tot}} = 1300$  p.u. and 2600 p.u.

optimal solution of the problem. Fig. 9 shows a linear relation between  $\log(\gamma)$  and  $B^{\text{tot}}$ .

Finally we study the effect of  $B^{\text{tot}}$  on the final configuration of battery locations and capacities. We compare the final configurations for  $B^{\text{tot}} = 1300$  p.u. and 2600 p.u. For both the cases we start from equally placing the batteries at the non-slack buses and start the optimization from a lower temperature  $T_c = 0.05$ . We do this because equal battery placement is already near the optimal solution. We repeat the optimization 10 times for different  $B^{\text{tot}}$  and take an average for presenting the final configurations in Fig. 10. Doubling  $B^{\text{tot}}$  does not affect the final configuration of the batteries, it almost doubles the size of batteries at the final configuration as shown in Fig. 10.

## 7 Conclusions

For finding the optimal storage placement to enhance power network reliability we use a novel combination of two computational techniques namely Simulated Annealing and the Splitting technique for rare-event simulation. To best of our knowledge this combination has not been used before. We use simulated annealing to minimize the reliability index, PLCV ( $\gamma$ ), of the network. We find that for very small values of  $\gamma$ , SA might not converge, however this problem disappears if we use  $\log(\gamma)$  instead of  $\gamma$  as the cost function in SA. In order to calculate the small values of  $\gamma$ 's we use FNS splitting technique for rare-event simulation.

We apply our method to the IEEE-14 bus network for three different examples. In example 1 we have different nodal power injections and line current maxima. For this example we find that at the final configuration (after the minimization), bus 3 has 35 percent of  $B^{\text{tot}}$ . In example 2 we keep the nodal power injections to be similar and find that at the final configuration buses 3, 8 and 10 each get 15 percent of  $B^{\text{tot}}$ .

For example 3 we keep all the nodal power injections and the line current maxima to be similar. For this we find that equal placement of the batteries at the non-slack nodes is the near optimal solution, which suggests that the number of connections to a node does affect the reliability of the power network. We also find that the  $\log(\gamma)$  decreases linearly with  $B^{\text{tot}}$ . Finally we examine if  $B^{\text{tot}}$  has an effect on final configuration of the battery locations and capacities. We observe that it does not affect the final configuration.

## Acknowledgments

This work is a part of the Industrial Partnership Program (IPP) 'Computational sciences for energy research' of the Foundation for Fundamental Research on Matter (FOM), which is financially supported by the Netherlands Organization for Research (NWO). This research program is co-financed by Shell Global Solutions International B.V.

## References

- [1] T. Yau, L. N. Walker, H. L. Graham, A. Gupta, and R. Raithel, "Effects of battery storage devices on power system dispatch," *Power Apparatus and Systems, IEEE Transactions on*, no. 1, pp. 375–383, 1981.
- [2] J. P. Barton and D. G. Infield, "Energy storage and its use with intermittent renewable energy," *Energy Conversion, IEEE Transactions on*, vol. 19, no. 2, pp. 441–448, 2004.
- [3] P. P. Varaiya, F. F. Wu, and J. W. Bialek, "Smart operation of smart grid: Risk-limiting dispatch," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 40–57, 2011.

- [4] M. Ghofrani, A. Arabali, M. Etezadi-Amoli, and M. S. Fadali, “A framework for optimal placement of energy storage units within a power system with high wind penetration,” *Sustainable Energy, IEEE Transactions on*, vol. 4, no. 2, pp. 434–442, 2013.
- [5] D. Gayme and U. Topcu, “Optimal power flow with large-scale storage integration,” *Power Systems, IEEE Transactions on*, vol. 28, no. 2, pp. 709–717, 2013.
- [6] E. Sjodin, D. F. Gayme, and U. Topcu, “Risk-mitigated optimal power flow for wind powered grids,” in *American Control Conference (ACC), 2012*, pp. 4431–4437, IEEE, 2012.
- [7] H. Aburub and W. T. Jewell, “Optimal generation planning to improve storage cost and system conditions,” in *PES General Meeting— Conference & Exposition, 2014 IEEE*, pp. 1–5, IEEE, 2014.
- [8] S. Bose, D. F. Gayme, U. Topcu, and K. M. Chandy, “Optimal placement of energy storage in the grid,” in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pp. 5605–5612, IEEE, 2012.
- [9] H. Oh, “Optimal planning to include storage devices in power systems,” *Power Systems, IEEE Transactions on*, vol. 26, no. 3, pp. 1118–1128, 2011.
- [10] K. M. Chandy, S. H. Low, U. Topcu, and H. Xu, “A simple optimal power flow model with energy storage,” in *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 1051–1057, IEEE, 2010.
- [11] J. van den Akker, S. Leemhuis, and G. Bloemhof, “Optimizing storage placement in electricity distribution networks,” in *Operations Research Proceedings 2012*, pp. 183–188, Springer, 2014.
- [12] W. Wadman, D. Crommelin, and J. Frank, “Applying a splitting technique to estimate electrical grid reliability,” in *Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World*, pp. 577–588, IEEE Press, 2013.
- [13] J. J. Grainger and W. D. Stevenson, *Power system analysis*, vol. 31. McGraw-Hill New York, 1994.
- [14] P. J. Van Laarhoven and E. H. Aarts, *Simulated annealing: theory and applications*, vol. 37. Springer Science & Business Media, 1987.
- [15] D. Bhaumik, D. Crommelin, and B. Zwart, “Mitigation of large power spills by a storage device.” <http://persistent-identifier.org/?identifier=urn:nbn:nl:ui:18-23525>, 2015. Submitted.
- [16] G. Rubino, B. Tuffin, *et al.*, *Rare event simulation using Monte Carlo methods*. Wiley Online Library, 2009.

- [17] M. Garvels, *The Splitting Method in Rare Event Simulation*. PhD thesis, Universiteit Twente, 2000.
- [18] M. Amrein and H. R. Künsch, “A variant of importance splitting for rare event estimation: Fixed number of successes,” *ACM Transactions on Modeling and Computer Simulation*, vol. 21, no. 2, p. 13, 2011.
- [19] M. J. Garvels, J.-K. C. Van Ommeren, and D. P. Kroese, “On the importance function in splitting simulation,” *European Transactions on Telecommunications*, vol. 13, no. 4, pp. 363–371, 2002.
- [20] W. Wadman, D. Crommelin, and B. Zwart, “A large deviation based splitting estimation of power flow reliability.” <http://homepages.cwi.nl/~wadman/work/WadmanCrommelinZwartPreprint.pdf#view=Fit>, 2015. Accepted for publication in TOMACS.
- [21] J. E. Freund and G. A. Simon, *Modern elementary statistics*, vol. 256. Prentice-Hall Englewood Cliffs, New Jersey, 1967.
- [22] 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,” *Power Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 12–19, 2011.