# Distributed Optimal Load Shedding for Disaster Recovery in Smart Electric Power Grids: A Second-Order Approach

Jia Liu[†] Cathy H. Xia[*] Ness B. Shroff[†] Hanif D. Sherali[‡]

[†]Department of Electrical and Computer Engineering, The Ohio State University

[*] Department of Integrated Systems Engineering, The Ohio State University

[‡]Grado Department of Industrial and Systems Engineering, Virginia Tech

**Abstract**

With the rapid growth of scale and complexity in modern electric power grids, conventional centralized load shedding schemes are being stretched to their limits when coping with disaster recovery (e.g., fires, quake, storms, terrorist attacks, etc.). In this paper, we consider the problem of distributed load shedding optimization for disaster recovery in smart grids. We develop distributed second-order interior-point based load shedding algorithms that enjoy a fast quadratic convergence rate. Contrary to existing first-order gradient based algorithms that are not only slow converging, but also could constantly violate the constraints, our interior-point based approach guarantees feasibility at all times – a highly desired feature to ensure stability and reliability for the power grid. Our main contributions are three-fold: (i) We propose a rooted spanning tree based reformulation that enables our distributed algorithm design; (ii) Based on the spanning tree reformulation, we propose a double Sherman-Morrison-Woodbury (SMW) scheme that yields distributed computation schemes for primal Newton directions and dual variables; (iii) We design an efficient scheme to initialize our second-order load shedding algorithm and propose a simplified step-size selection strategy, which is well suited for implementations in practice and offers near-optimality performance guarantee. Collectively, these results serve as an important first step in load shedding and disaster recovery that uses second-order distributed techniques.

## 1 Introduction

As modern electric power grids evolve into increasingly complex networked systems, it becomes even more important that they should be self-sustaining and self-healing after extreme disasters strike (e.g., quakes, storms, fires, terrorist attacks, etc.). In post-disaster recovery, due to the significant imbalance between power demand and supply as well as large deficits/stresses on the transmission networks, the load must be *rapidly* dropped to prevent expensive damages to the power systems until the completion of infrastructure repairs, which often takes days or even weeks to finish. On the other hand, post-disaster load shedding also needs to be performed in an intelligent and selective way to *minimize* the impact of large-scale blackouts, which could lead not only to wide-spread inconvenience, but also result in huge economic losses and even loss of lives. For example, priority should be given to major hospitals/health care institutions, sewerage and water pumping stations, industries requiring continuous power supply, and major public transportation such as traffic lights, train stations, airports, etc. However, due to the constantly growing scale and complexity of power grids, traditional centralized load shedding schemes are having difficulties in coping with single-point-of-failure and/or islanding types of problems resulting from disastrous events, risking the entire power system being unstable and vulnerable to a complete shut-down (i.e., "cascading failure," recent examples include the major power outage in northeast America and Canada in 2003 [1]). The limitations of centralized load shedding will be even more pronounced in future smart grids that contain a large number of unplanned and distributed renewable energy generations.

To address the above challenges, in this paper, we consider designing a *distributed interior-point based second-order* load shedding algorithm. The salient features of our approach are: (i) Our algorithm, being an interior-point method, can guarantee that the operating point of the electric power grid stays *feasible* throughout all iterations. This is a highly desirable feature for ensuring stable and reliable operations of the power system. In contrast, most first-order gradient based search schemes have no such feasibility guarantee and would produce iterates that constantly violate the feasibility region. (ii) By considering the second-order Hessian information in distributed load shedding, we can expect, as in classical nonlinear optimization theory [2], to alleviate the inherent ill-conditioned limitation of most conventional first-order gradient-based load shedding optimization schemes (see Section 2 for detailed discussions), thus leading to a much faster convergence that is crucial in load shedding.

However, developing distributed second-order load shedding algorithms is highly challenging and results in this area remain elusive. Due to the unique nature of power flow equations and the very different problem structure in power systems, techniques used for distributed second-order algorithms in wireline/wireless communications networks [3–6] cannot be directly applied. Generally speaking, to design a distributed second-order algorithm for load shedding optimization, one typically needs to compute the primal Newton and dual search directions, which in turn requires decomposing the inverses of the Hessian matrix and a weighted Laplacian matrix, and then distributing each piece to the according network entity (i.e., a bus or a transmission line). Unfortunately, unlike communication networks for which the Hessian matrix has a nice sparsity structure (e.g., diagonal [4], block diagonal [5], "arrow-headed" [6]), the Hessian structure of the load shedding problem is *non-separable* because of the line flows that are governed by the Kirchoff's law. Moreover, not only are both the Hessian and weighted Laplacian inversions cumbersome to compute in large-scale power networks, the obtained inverses also have little sparsity structure in general. As a result, designing distributed second-order algorithms are far more difficult for power systems, which contain both cyber and physical components.

The key contribution of this paper is the development of a set of new second-order techniques to address all the aforementioned technical difficulties in distributed load shedding. The main results and technical contributions of this paper are as follows:

- We first propose a rooted spanning tree based problem reformulation that not only reduces the size of the problem, but also enables the uses of advanced distributed spanning tree networking protocols for message passing in power systems – a highly desirable feature in future smart grids. Further, we show how one can exploit the special structure of the inverse incidence matrix of the rooted spanning tree to give rise to distributed second-order load shedding algorithm design.

- Based on the rooted spanning tree reformulation, we propose a double Sherman-Morrison-Woodbury (SMW) approach that yields distributed computation schemes for primal Newton directions and dual variables. For a power network with $N$ buses and $L$ lines, we show that the double SMW approach obtains the *precise* solutions of Hessian and Laplacian inverse in $L$ and $N+1$ steps, respectively, rather than asymptotically as in [4–6]. This makes our proposed algorithm a fully second-order approach without the assumption of time-scale separation. It also significantly enhances the practicality of our second-order method in electric power grids.

- We design an efficient scheme to initialize our second-order load shedding algorithm and design a relaxed step-size selection strategy, which is well-suited for distributed implementations in practice and offers near optimality performance guarantee. We offer new insights and networking interpretations for our proposed distributed second-order load shedding algorithms, as well as the connections with and differences from first-order approaches. This further advances our understanding of second-order approaches in power systems optimization theory.

Collectively, these results serve as an important first step in load shedding and disaster recovery that uses second-order distributed techniques. The remainder of this paper is organized as follows. In Section 2, we review related works. Section 3 introduces the network model and problem formulations. Section 4 presents the key components of our distributed load shedding schemes. Section 5 presents numerical results and Section 6 concludes the paper.

## 2 Related Work

Load shedding optimization lies at the heart of several important related areas in power grids studies, such as vulnerability and contingency analysis (see, e.g., [7–9]), cascading failure analysis [10–12], etc.

To our knowledge, initial attempts on load shedding optimization appeared as early as in late '60s, when Hajdu *et al.* [13] developed centralized gradient techniques for load shedding. Because of its importance in electric power grids, load shedding optimization continues to be an actively research area today (see, e.g., [12,14,15] and references therein). However, the predominant approach so far in this area is still first-order gradient based, which suffers from the long convergence time issue and, as mentioned earlier, have no feasibility guarantee during their convergence processes. Moreover, existing works in this area are mostly centralized, which are unsuitable for post-disaster recovery that calls for *fast-response* and *distributed* schemes. On the other hand, most works on distributed load shedding to date are based on heuristics, meta-heuristics, and the knowledge-based systems [16–20], which do not offer performance guarantees. In contrast, in this paper, we seek to design fast and distributed load shedding algorithms based on rigorous interior-point optimization theory that provides optimal and fast convergence performances.

Our work is also related to the recent studies on distributed second-order methods for communications and networking systems [3–6]. However, due to the unique cyber-physical nature of electric power networks, techniques developed in these works are not directly applicable here. In particular, we note that the matrix-splitting scheme in [3–6], a key technique used for distributed Hessian and Laplacian matrix inversions, is actually a first-order method and has a slow convergence speed (see numerical studies in Section 5), which is impractical for load shedding under disasters. To overcome these limitations, in this paper, we develop a new double SMW procedure for Hessian and Laplacian matrix inversions, which efficiently determines precise solutions. This is in stark contrast to those matrix-splitting based schemes in [3–6] that only converge asymptotically.

# 3 Problem Formulation

## 3.1 System Model and Problem Formulation

We first introduce the notation style in this paper. We use boldface to denote matrices and vectors. We let $\mathbf{A}^T$ denote the transpose of $\mathbf{A}$. Diag $\{\mathbf{A}_1, \ldots, \mathbf{A}_N\}$ represents the block diagonal matrix with $\mathbf{A}_1, \ldots, \mathbf{A}_N$ on its main diagonal. We let $(\mathbf{A})_{ij}$ represent the entry in the $i$-th row and $j$-th column of $\mathbf{A}$. We let $(\mathbf{A})_m$ and $(\mathbf{v})_m$ represent the $m$-th column and $m$-th entry of $\mathbf{A}$ and $\mathbf{v}$, respectively. We let $\mathbf{I}_K$ denote the $K$-dimensional identity matrix, and let $\mathbf{1}_K$ and $\mathbf{0}_K$ denote the $K$-dimensional vectors whose elements are all ones and zeros ("$K$" may be omitted for brevity if the dimension is clear from the context). We let $\mathbf{e}_K^{(k)}$ denote the $k$-th vector in the natural basis of $\mathbb{R}^K$ (i.e., the $k$-th entry is "1" and other entries are "0"). We let $\lambda_{\min}\{\mathbf{A}\}$ and $\lambda_{\max}\{\mathbf{A}\}$ denote the smallest and largest eigenvalues of $\mathbf{A}$, respectively.

**a) The Power Grid Model:** Consider a power grid with failed generators and transmission lines after a diaster strikes such that it requires load shedding to avoid potentially large-scale cascading failures. Without loss of generality, we consider a power (sub)network that remains connected upon removal of the failed portion of the system, because otherwise we can perform load shedding on each connected component individually. However, to model post-disaster recovery, we do *not* assume the availability of *any* centralized controller in *any* sub-network. We represent the network by a connected directed graph $\mathcal{G} = \{\mathcal{N}, \mathcal{L}\}$, where $\mathcal{N}$ and $\mathcal{L}$ are the sets of buses (i.e., nodes) and transmission lines (i.e., links), with $|\mathcal{N}| = N$ and $|\mathcal{L}| = L$, respectively. We use a *node-arc incidence matrix* (NAIM) [21] $\mathbf{A} \in \mathbb{R}^{N \times L}$ to represent the network topology. Let $\mathrm{Tx}(l)$ and $\mathrm{Rx}(l)$ denote the transmitting and receiving buses of line $l$, respectively. The entries $(\mathbf{A})_{nl}$ are defined as follows:

$$(\mathbf{A})_{nl} = \begin{cases} 1 & \text{if } n = \mathrm{Tx}(l), \\ -1 & \text{if } n = \mathrm{Rx}(l), \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

We suppose that there are $G$ generator buses in this power network and the remaining $N - G$ buses are load buses. We let $\mathcal{K}$ denote the set of generators, with $|\mathcal{K}| = G$.

We assume that time is slotted and that each time slot is sufficiently long so that steady-state stable operating point analysis is reasonable (e.g., 1 minute [22]). In this paper, we do not consider the current and voltage transients ensuing load shedding, which quickly die away (typically lasting a few system cycles). Instead, we refer interested readers to [23] for details in transient stability analysis. In time slot $t$, we let $p_n[t]$ denote the random power demand at a load bus $n \notin \mathcal{K}$ in time slot $t$. The stochastic demand process $\{p_n[t]\}$ is assumed to be ergodic. Here, following the convention in power system literature, we set $p_n[t]$ to be non-positive (i.e., $p_n[t] \leq 0$) to signify that it represents power consumption rather than generation. For convenience, we use a vector $\mathbf{p}^l_{[t]} \triangleq [p_{G+1}[t], \ldots, p_N[t]]^T \in \mathbb{R}^{N-G}_{-}$ to group the stochastic power demands at all load buses in time slot $t$. As in standard in contingency analysis, we assume that the load of each generator bus is negligible. Therefore, we let $\mathbf{p}_{[t]} \triangleq [\mathbf{0}_G^T, (\mathbf{p}^l_{[t]})^T]^T \in \mathbb{R}^N_{-}$ denote the expanded power demand vector at all buses in the power grid.

**b) The Load Shedding Model:** We let $z_n[t] \geq 0$ denote the amount of power to be shed at load bus $n \notin \mathcal{K}$ in time slot $t$. Accordingly, we let $z_n[t] \geq 0$, $n \in \mathcal{K}$, denote the corresponding power generation at each generator to meet the demands after load-shedding. For convenience, we use $\mathbf{z}_{[t]} \triangleq \left[z_1[t], \ldots, z_N[t]\right]^T \in \mathbb{R}^N_{+}$ to group all power generation and load-shedding variables in time slot $t$. As in standard contingency analysis, we assume that the active power losses in transmission lines are negligible (i.e., lines are purely reactive). Then, from the active power conservation law, we have $\sum_{n\in\mathcal{K}} z_n[t] + \sum_{n\notin\mathcal{K}}(p_n[t] + z_n[t]) = 0$, $\forall t$, which can be further written in vector form as:

$$\mathbf{1}^T(\mathbf{p}_{[t]} + \mathbf{z}_{[t]}) = 0, \quad \forall t. \tag{2}$$

It is natural to require that

$$p_n[t] + z_n[t] \leq 0, \quad n \notin \mathcal{K}, \tag{3}$$

which means that we will not shed more than the actual demand. Also, to model the output limit of generators, we have

$$z_n[t] \leq B_n^{\max}, \quad n \in \mathcal{K}, \tag{4}$$

where $B_n^{\max} > 0$ represents the capacity limit of generator $n$.

We let $f_l[t]$ represent the electric power flow on line $l$ in time slot $t$ under load-shedding. We allow $f_l[t]$ to be negative in the sense that when $f_l[t] < 0$, it simply means that the direction of the power flow is against the direction of the transmission line. Then, the power flow conservation law in the electric power grid can be expressed as:

$$\sum_{l\in\mathcal{I}(n)} f_l[t] = (p_n[t] + z_n[t]) + \sum_{l\in\mathcal{O}(n)} f_l[t], \quad \forall n,$$

where $\mathcal{I}(n)$ and $\mathcal{O}(n)$ represent the incoming and outgoing lines at bus $n$, respectively.

For convenience, we use $\mathbf{f}_{[t]} \triangleq [f_1[t], \ldots, f_L[t]]^T \in \mathbb{R}^L$ to group all line flow variables in time slot $t$. With the node-arc-incidence matrix $\mathbf{A}$, it can be readily verified that the power flow conservation law can be compactly written as:

$$\mathbf{A}\mathbf{f}_{[t]} - \mathbf{z}_{[t]} = \mathbf{p}_{[t]}. \tag{5}$$

Further, since all rows of $\mathbf{A}$ sum up to zero, i.e., $\mathbf{1}^T\mathbf{A} = \mathbf{0}^T$, we have $\mathbf{1}^T(\mathbf{p}_{[t]} + \mathbf{z}_{[t]}) = \mathbf{1}^T\mathbf{A}\mathbf{f}_{[t]} = (\mathbf{1}^T\mathbf{A})\mathbf{f}_{[t]} = \mathbf{0}$, which means that (5) implies (2). As a result, there is no need to explicitly list (2) in our load shedding model. An important goal in load shedding and the associated power flow distribution is to prevent overloaded transmission lines. Let $f_l^{\max}$ denote the upper limit of the line flow for line $l$, beyond which a line outage is likely to happen. Then, the line outage constraint can be written as:

$$-f_l^{\max} \leq f_l[t] \leq f_l^{\max}, \quad \forall l, t. \tag{6}$$

As in the contingency analysis literature [7–9], we assume that the voltage magnitudes at all buses are fixed at a nominal value and hence the real power flow distributions on all transmission line are controlled by the phase angles of all buses, making the reactive power constraint unnecessary (see [22, Chap. 16.4] for more details of this model). Let $\theta_n[t]$, $n = 1, \ldots, N$, denote the operation phase angle of bus $n$ in time slot $t$. We use $\boldsymbol{\theta}_{[t]} \triangleq [\theta_1[t], \ldots, \theta_N[t]]^T \in \mathbb{R}^N$ to group all phase angle variables. We let $b_l \in \mathbb{R}_+$ represent the admittance of line $l$. We further define a diagonal line admittance matrix $\mathbf{B} \triangleq \operatorname{Diag}\{b_1, \ldots, b_L\} \in \mathbb{R}^{L \times L}$. Then, the (DC) line flow model can be written as:

$$\mathbf{f}_{[t]} = \mathbf{B}\mathbf{A}^T \boldsymbol{\theta}_{[t]}, \tag{7}$$

$$-\beta \mathbf{1} \leq \mathbf{A}^T \boldsymbol{\theta}_{[t]} \leq \beta \mathbf{1}, \quad \forall t, \tag{8}$$

where $\beta \in (0, \frac{\pi}{2})$ is some appropriate bounding constant. Here, note that the relationship between phase angles, admittance, and power flow in (7) is identical to that between voltages, conductance, and current under the Kirchoff's law, hence the name DC model. We note that the DC model in (7) and (8) is an approximation of the exact line flow $\mathbf{f}_{[t]} = \mathbf{B}\sin(\mathbf{A}^T\boldsymbol{\theta}_{[t]})$, $-\frac{\pi}{2}\mathbf{1} \leq \mathbf{A}^T\boldsymbol{\theta}_{[t]} \leq \frac{\pi}{2}\mathbf{1}$ in AC systems ($\sin(\mathbf{A}^T\boldsymbol{\theta}_{[t]}) \in \mathbb{R}^L$ is a vector whose entries are $\sin((\mathbf{A}^T\boldsymbol{\theta}_{[t]})_l)$, $\forall l$). This is based on the fact that when $|\mathbf{A}^T\boldsymbol{\theta}_{[t]}|$ is small, we have $\sin(\mathbf{A}^T\boldsymbol{\theta}_{[t]}) \approx \mathbf{A}^T\boldsymbol{\theta}_{[t]}$. We note, however, that for disastrous events that necessitate load shedding, the DC model is more preferable over the AC model [22]. This is because great precision is not of high priority in such cases, and the primary interest is to quickly cope with insecure or vulnerable conditions following any outages. The DC model alleviates the non-convex difficulty of the AC model and yields a more computationally tractable model. However, due to the coupled problem structure as we show later, it remains a very challenging task to design distributed load shedding algorithm based on the DC model.

It is worth pointing out that, rather than just being an approximation of the AC model, the DC model has theoretical and practical significance in its own right. In practice, it has been customary to prescribe very conservative $\beta$-values so that other state variables (e.g., rates of change of angle differences $d(\theta_{\mathrm{Tx}(l)} - \theta_{\mathrm{Rx}(l)})/dt$) are unlikely to cause instability [13].

**c) Objective function:** In this paper, our goal is to minimize the blackout impacts resulting from load shedding in disaster recovery. The blackout impacts due to load shedding in diaster recovery can be characterized as:

$$\sum_{n \notin \mathcal{K}} C_n \left( \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} z_n[t] \right), \tag{9}$$

where $C_n(\cdot)$ is a cost function that is usually assumed to be strictly convex and twice-differentiable in power system analysis [24]. For example, $C_n(\cdot)$ could be quadratic [25] with a non-negative weight, which represents different priorities given to different loads in disaster recovery (e.g., higher priorities are given to hospitals, water systems, transportation, etc.). We remark that in (9), $T \to \infty$ models the long and uncertain period of time (e.g., days or weeks) needed for infrastructure repairs in disaster recovery relative to the short time slot unit in load shedding decisions.

Putting together all the modeling in a)–c) yields the following load shedding optimization problem

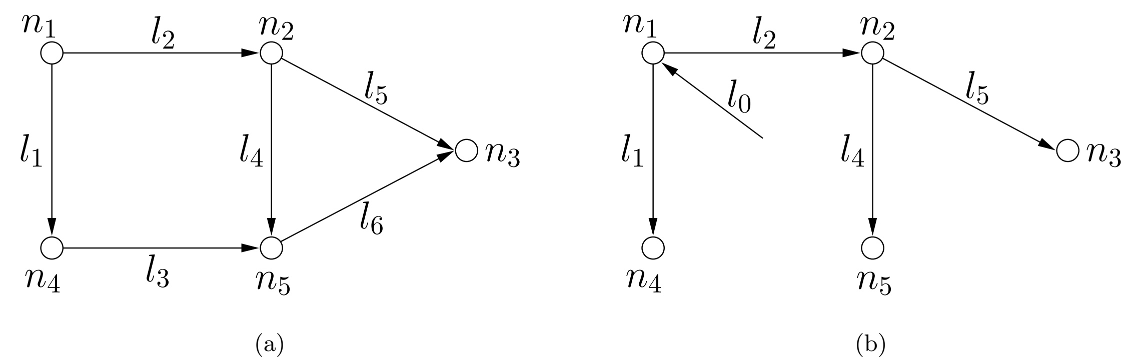

Figure 1: An illustrative example of constructing a rooted spanning tree in an electric power grid: (a) the original network, (b) a rooted spanning tree with an artificial arc incident upon the root at bus $n_1$.

(LSO):

**LSO:**

| | |
|---|---|
| Minimize | Objective in (9) |
| subject to | Constraints in $(3),(4),(5),(6),(7),$ and $(8)$. |

### 3.2 A Rooted Spanning Tree Based Reformulation

To solve Problem LSO, we first consider the following static optimization problem, where we drop the time index "$[t]$" for now and all decision variables represent the long-term averages:

**LSO-S:**

$$\begin{aligned}
&\text{Minimize} && \sum_{n \notin \mathcal{K}} C_n\left(z_n\right) \\
&\text{subject to} && \mathbf{A}\mathbf{f} - \mathbf{z} = \mathbf{p}, && \text{(10a)} \\
& && \mathbf{f} - \mathbf{B}\mathbf{A}^T\boldsymbol{\theta} = \mathbf{0}, && \text{(10b)} \\
& && -\beta\mathbf{1} \le \mathbf{A}^T\boldsymbol{\theta} \le \beta\mathbf{1}, && \text{(10c)} \\
& && -f_l^{\max} \le f_l \le f_l^{\max},\ \forall l, && \text{(10d)} \\
& && 0 \le z_n \le K_n,\ \forall n, && \text{(10e)}
\end{aligned}$$

where, for convenience, we define $K_n \triangleq B_n^{\max}$ for $n \in \mathcal{K}$ and $K_n \triangleq -p_n$ for $n \notin \mathcal{K}$. Note that since $rank(\mathbf{A}) = N - 1$ [21], the linear equation systems in (10a) and (10b) are rank-deficient: Eq. (10a) is not full row-rank and (10b) has more rows than varaibles. Therefore, in order for (10a) and (10b) to be consistent (i.e., having solutions), one needs to resolve the rank-deficiency issue.

Toward this end, we propose a *rooted spanning tree* approach inspired by the network simplex algorithm for solving linear max-flow problems [21]. Without loss of generality, we assume that a generator bus is always labeled as bus 1 and we designate this bus to be the root node. Then, one can add an artificial root arc (line) $l_0$ to the root node (bus), and the root arc has arbitrary admittance $b_0$ with an artificial line flow $f_0$ originating from space and terminating at bus 1 (see Figure 1 for an illustrative example)[1]. Adding an artificial line flow is equivalent to adding a linearly independent column to to (10a) to make it full row-rank. Moreover, it is not difficult to verify that as long as the active power conservation in (2) holds, we have $f_{(0)} \equiv 0$, implying that the resultant system is equivalent.

Next, we let $\mathcal{T} = \{\mathcal{N}, \mathcal{L}' \cup \{l_0\}\}$ be a spanning tree rooted at bus 1, where the link set $\mathcal{L}' \subseteq \mathcal{L}$ and $|\mathcal{L}'| = N - 1$. Pruning the network into a spanning tree corresponds to removing the redundant rows in (10b). We let $\widetilde{\mathbf{A}} \triangleq [-\mathbf{e}_N^{(1)} \; (\mathbf{A})_{l\in\mathcal{L}'}] \in \mathbb{R}^{N\times N}$ be the node-arc incidence matrix (NAIM) of the rooted spanning tree. For convenience, we let $\mathcal{L}_0' \triangleq \mathcal{L}' \cup \{l_0\}$ denote all links of the rooted spanning tree $\mathcal{T}$. Accordingly, we let vector $\widetilde{\mathbf{f}} \triangleq [f_l : l \in \mathcal{L}_0']^T \in \mathbb{R}^N$ group all the line flows in the rooted spanning tree. We let $\widetilde{\mathbf{B}} \triangleq \mathrm{Diag}\,\{b_l : l \in \mathcal{L}_0'\} \in \mathbb{R}^{N\times N}$ be the diagonal line admittance matrix with respect to $\mathcal{T}$. Next, replacing $\mathbf{A}$, $\mathbf{f}$, and $\mathbf{B}$ in (10a) and (10b) with $\widetilde{\mathbf{A}}$, $\widetilde{\mathbf{f}}$, and $\widetilde{\mathbf{B}}$, respectively; substituting (10b) into (10a); and then combining (10c) and (10d) by noting $\mathbf{f} = \mathbf{B}\mathbf{A}^T\boldsymbol{\theta}$, we arrive at the following rooted spanning tree based problem formulation:

**LSO-RST:**

$$\begin{aligned}
&\text{Minimize} && \sum_{n\notin\mathcal{K}} C_n(z_n) && \\
&\text{subject to} && (\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^T)\boldsymbol{\theta} - \mathbf{z} = \mathbf{p}, && \text{(11a)}\\
& && \mathbf{1}^T(\mathbf{p}+\mathbf{z}) = 0, && \text{(11b)}\\
& && -\beta_l \le \theta_{\mathrm{Tx}(l)} - \theta_{\mathrm{Rx}(l)} \le \beta_l, \quad \forall l, && \text{(11c)}\\
& && 0 \le z_n \le K_n, \quad \forall n, && \text{(11d)}
\end{aligned}$$

where we explicitly expand $\mathbf{A}^T\boldsymbol{\theta}$ in scalar form and let $\beta_l \triangleq \min\{\beta, f_l^{\max}/b_l\}$ in (11c). The added constraint in (11b) is to enforce that the artificial line flow $f_0 \equiv 0$.

It is not difficult to show that Problem LSO-RST is equivalent to Problem LSO-S. In other words, if LSO-S has a solution, then it must be a solution to LSO-RST, and vice versa. Thus, it suffices to solve LSO-RST. More importantly, we will see in later sections that the rooted-spanning tree reformulation enables us to derive a distributed solution.

Several remarks on the rooted spanning tree reformulation are in order. First, the rooted spanning tree reformulation eliminates the line flow variables $\mathbf{f}$ and the $O(N^2)$ constraints in (10b), hence significantly reducing the problem size. Second, the root node is not just an abstract mathematical

[1] In [21], the rooted spanning tree is constructed in a different but equivalent way: the root is chosen among the sink nodes and the root arc originates from the root and terminates in space.

construct, but also has a physical meaning: it corresponds to the *slack bus* in power systems, which balances the overall power production. Although in theory any generator can be selected as the root, for power systems in reality, it is more preferable to choose a generator that has the largest output limit. Third, although in general the rooted spanning tree is not unique, the specific choice of rooted spanning tree does not affect our subsequent algorithm design. But from performance perspective, it is more advantageous to use a spanning tree with a small network diameter (see discussions in Section 4.2). Lastly, although the rooted spanning tree reformulation is inspired by the network simplex algorithm, Problem LSO-RST has a nonlinear objective function and a far more complex coefficient matrix $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^T$, which can *not* be handled by the network simplex algorithm.

Since identifying spanning trees is a mature subject, its details are beyond the scope of this work. We instead remark here that a spanning tree can be efficiently identified in a *distributed* fashion and the state-of-the-art complexity is $O(\sqrt{N}\log(N))$ [26]. In this work, we assume that after the disaster, a new rooted spanning tree can be re-generated within the connected sub-network by a *distributed* spanning tree algorithm (recall that we do not assume any centralized controller in any sub-network). Further, as in standard outputs of all distributed spanning tree algorithms, every bus knows which lines belong to the rooted spanning tree and which do not (i.e, the knowledge of $\widetilde{\mathbf{A}}$). This assumption is reasonable since the topology of an electric power grid can be considered stationary in the convergence time scale of spanning tree algorithms.

Further, due to the special structure of $\widetilde{\mathbf{A}}$, every bus can permute the rows and columns in $\widetilde{\mathbf{A}}$ in such a way that $\widetilde{\mathbf{A}}$ is *upper triangular* with only $\pm 1$ and 0 elements [21]. As an example, for the network in Figure 1(a), the rearranged upper triangular NAIM of the rooted spanning tree in Figure 1(b) is as follows:

$$\widetilde{\mathbf{A}} = \begin{array}{c|c:cccc} & l_0 & l_2 & l_5 & l_1 & l_4 \\ \hline n_1 & -1 & 1 & 0 & 1 & 0 \\ n_2 & 0 & -1 & 1 & 0 & 1 \\ n_3 & 0 & 0 & -1 & 0 & 0 \\ n_4 & 0 & 0 & 0 & -1 & 0 \\ n_5 & 0 & 0 & 0 & 0 & -1 \end{array}.$$

Next, we introduce a sparsity result about $\widetilde{\mathbf{A}}^{-1}$ [21, Page 430], which will be useful in our efficient distributed algorithm designs later.

**Lemma 1** (Inverse of $\widetilde{\mathbf{A}}$)**.** *Let $\widetilde{\mathbf{A}}$ be the NAIM of a rooted spanning tree that is arranged in an upper triangular form. Then, $\widetilde{\mathbf{A}}^{-1}$ is also an upper triangular matrix with only $\pm 1$ and 0 elements.*

## 4 A Distributed Second-Order Load Shedding Algorithm

A common approach to solve Problem LSO is to accommodate the constraints (3)–(8) into the objective function by using Lagrangian dual variables (see, e.g., [12,14,15]), and then use the standard (sub)gradient (steepest descent) method to determine the optimal dual variables in the dual problem [2]. However, the solution to the dual problem typically approaches optimality from the exterior of the feasible domain. That is, the constraints (3)–(8) could be constantly violated during the process. This may not be a big issue for communications networks (thanks to buffering and delay tolerance), but is *not* acceptable for power grids as it could lead to serious reliability and stability concerns. It is also well-known that gradient based methods suffer from a slow convergence rate due to its first-order nature. These limitations motivate us to develop an interior-point based distributed second-order approach. The most attractive feature of the interior-point approach is that it guarantees feasibility in all iterates, which is critical in power grids load shedding. In Sections 4.1 and 4.2, we present our interior-point based second-order optimization algorithm and its distributed design, respectively. In Section 4.3, we discuss strategies that can be used to quickly initialize our second-order algorithm. Finally in Section 4.4, we show a simple step-size selection strategy that is well suited for practical implementation and yet maintain the optimality guarantee of the algorithm in terms of average load shedding.

### 4.1 An interior-point second-order approach

Following the standard interior-point approach [27], we apply a logarithmic barrier function to all inequality constraints in Problem LSO-RST and then accommodate them into the objective function. The resultant augmented objective function can be written as:

$$
\begin{aligned}
g_\mu(\mathbf{y}) \triangleq \mu \sum_{n \notin \mathcal{K}} C_n(z_n) - \sum_{n=1}^{N} (\log(z_n) + \log(K_n - z_n)) \\
- \sum_{l=1}^{L} (\log(\beta_l - \theta_{\mathrm{Tx}(l)} + \theta_{\mathrm{Rx}(l)}) + \log(\beta_l + \theta_{\mathrm{Tx}(l)} - \theta_{\mathrm{Rx}(l)})),
\end{aligned}
\tag{12}
$$

where $\mathbf{y} \triangleq \left[z_1 \cdots z_N, \theta_1 \cdots \theta_N\right]^T \in \mathbb{R}^{2N}$ groups all variables. In (12), $\mu > 0$ is a parameter that is used to track the central path in the interior-point method as $\mu \to \infty$ [27]. Moreover, we let $\mathbf{M} \triangleq \left[\begin{array}{c:c} -\mathbf{I} & \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^T \\ \hdashline -\mathbf{1}_N^T & \mathbf{0}_N^T \end{array}\right] \in \mathbb{R}^{(N+1)\times 2N}$ and $\mathbf{d} \triangleq [\mathbf{p},\ \mathbf{1}^T\mathbf{p}]^T \in \mathbb{R}^{N+1}$. Then, we obtain the following $\mu$-scaled approximating problem of Problem LSO-RST:

$$
\begin{aligned}
\boldsymbol{\mu}\textbf{-LSO-RST:}\ \text{Minimize} \quad & g_\mu(\mathbf{y}) \\
\text{subject to} \quad & \mathbf{M}\mathbf{y} = \mathbf{d},
\end{aligned}
\tag{13}
$$

In $g_\mu(\mathbf{y})$, note that as $\mu \to \infty$, the original objective function of Problem LSO-RST dominates the barrier functions, and hence the solution to Problem (13) approaches that of Problem LSO-RST asymptotically. Further, since $\mu$ can be increased exponentially (e.g., letting $\mu_k = 2^k$), it suffices to focus on a second-order solution to the $f_\mu(\mathbf{y})$ problem in order to achieve a second-order convergence speed.

Next, we solve Problem (13) by the (centralized) Newton's method, which is a *second-order* algorithm. Starting from an initial feasible solution $\mathbf{y}^0$, the centralized Newton's method iteratively searches for an optimal solution as follows:

$$\mathbf{y}^{k+1} = \mathbf{y}^k + s^k \Delta \mathbf{y}^k, \tag{14}$$

where $s^k > 0$ is a positive step-size in the $k$-th iteration. In (14), $\Delta\mathbf{y}^k$ denotes the primal Newton direction, which is the solution to the following linear equation system obtained by deriving the Karush-Kuhn-Tucker (KKT) system of the second-order approximation of $g_\mu(\mathbf{y})$ [2]:

$$\begin{bmatrix} \mathbf{H}_k & \mathbf{M}^T \\ \mathbf{M} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{y}^k \\ \mathbf{w}^k \end{bmatrix} = - \begin{bmatrix} \nabla g_\mu(\mathbf{y}^k) \\ \mathbf{0} \end{bmatrix}, \tag{15}$$

where $\mathbf{H}_k \triangleq \nabla^2 g_\mu(\mathbf{y}^k) \in \mathbb{R}^{2N\times 2N}$ is the Hessian matrix of $g_\mu(\mathbf{y})$ at $\mathbf{y}^k$, and the vector $\mathbf{w}^k \in \mathbb{R}^{N+1}$ contains the dual variables with entries arranged as $[w_1^k, \ldots, w_{N+1}^k]^T$, where $w_n^k$ is the $k$-th iteration value of the dual variable associated with the $n$-th row in $\mathbf{My} = \mathbf{d}$. It can be readily verified that the coefficient matrix of the linear equation in (15) is nonsingular. Therefore, the primal Newton direction $\Delta\mathbf{y}^k$ and the dual variables $\mathbf{w}^k$ can be uniquely determined by solving (15). Further, with an appropriate step-size selection (to be discussed in more detail in Section 4.4), the second-order convergence speed of the above second-order scheme follows from standard interior-point methods [27].

However, we note that the above second-order approach creates many challenges for a distributed algorithm design since solving for $\Delta\mathbf{y}^k$ and $\mathbf{w}^k$ simultaneously via (15) requires global information. In what follows, we focus on designing distributed computation schemes for $\Delta\mathbf{y}^k$ and $\mathbf{w}^k$.

### 4.2 Distributed Algorithm Design

Our first step towards solving (15) in a decentralized manner is to derive a reduced linear system of (15) by Gaussian elimination as follows:

$$\Delta\mathbf{y}^k = -\mathbf{H}_k^{-1}(\nabla g_\mu(\mathbf{y}^k) + \mathbf{M}^T\mathbf{w}^k), \tag{16}$$

$$\mathbf{w}^k = (\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T)^{-1}(-\mathbf{M}\mathbf{H}_k^{-1}\nabla g_\mu(\mathbf{y}^k)). \tag{17}$$

Thus, at a given $\mathbf{y}^k$, we can solve for $\mathbf{w}^k$ from (17), which can be used to solve for $\Delta\mathbf{y}^k$ from (16). Then, the next iterate $\mathbf{y}^{k+1}$ can computed by (14) along with an appropriate step-size $s^k$. However, as we shall see shortly, the distributed computations of $\mathbf{H}_k^{-1}$ and $(\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T)^{-1}$ (which can be viewed as a Laplacian-type matrix [28] weighted by $\mathbf{H}_k^{-1}$) remain difficult due to the *non-separable* structures in $\mathbf{H}_k$ and $\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T$. This is in stark contrast to those problems in communications networks [3–5], where the Hessian matrices are (block) diagonal and their distributed inverse computations are much easier. In what follows, we will further exploit the special structure of $\mathbf{H}_k$ and $\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T$ in power systems to design a new iterative scheme to compute $\Delta\mathbf{y}^k$ and $\mathbf{w}^k$ in a distributed fashion.

**a) Distributed computation of primal Newton directions.** For simplicity, here we introduce several notations. We let $\Psi(n) \triangleq \mathcal{I}(n) \cup \mathcal{O}(n)$ denote the set of lines that incident upon bus $n$. If bus $n$ is one of the two end buses of line $l$, we use the notation $\mathcal{E}(l)\backslash n$ to represent the other end bus of line $l$ (where $\mathcal{E}(l)$ denotes the set of two end buses of $l$). Also, we let $C_n(\cdot) \equiv 0$ for $n \in \mathcal{K}$.

To see the non-separable structure of $\mathbf{H}_k$, we evaluate the first and second partial derivatives of $g_\mu(\mathbf{y})$, for which the non-zero ones are:

$$\frac{\partial g_\mu}{\partial z_n} = \mu C_n'(z_n) + \frac{1}{K_n - z_n} - \frac{1}{z_n}, \quad \forall n,$$
$$\frac{\partial^2 g_\mu}{\partial z_n^2} = \mu C_n''(z_n) + \frac{1}{(K_n - z_n)^2} + \frac{1}{z_n^2}, \quad \forall n,$$
$$\frac{\partial g_\mu}{\partial \theta_n} = \sum_{l \in \Psi(n)} \frac{2(\theta_n - \theta_{\mathcal{E}(l)\backslash n})}{\beta_l^2 - (\theta_n - \theta_{\mathcal{E}(l)\backslash n})^2}, \quad \forall n,$$
$$\frac{\partial^2 g_\mu}{\partial \theta_n \partial \theta_{n'}} = \begin{cases} \sum_{l \in \Psi(n)} \frac{2\left[\beta_l^2 + (\theta_n - \theta_{\mathcal{E}(l)\backslash n})^2\right]}{\left[\beta_l^2 - (\theta_n - \theta_{\mathcal{E}(l)\backslash n})^2\right]^2}, & \text{if } n = n', \\ -\frac{2\left[\beta_l^2 + (\theta_n - \theta_{n'})^2\right]}{\left[\beta_l^2 - (\theta_n - \theta_{n'})^2\right]^2}, & \text{if } n, n' \text{ is connected by } l. \end{cases}$$

We further define a diagonal matrix $\mathbf{Z} \triangleq \mathrm{Diag}\{\frac{\partial g_\mu^2}{\partial z_n^2}, n = 1, \ldots, N\} \in \mathbb{R}^{N \times N}$ and a matrix $\mathbf{\Theta} \in \mathbb{R}^{N \times N}$ with entries $(\mathbf{\Theta})_{nn'} = \frac{\partial^2 g_\mu}{\partial \theta_n \partial \theta_{n'}}$. Then, $\mathbf{H}_k$ can be written in a block diagonal form: $\mathbf{H}_k = \mathrm{Diag}\{\mathbf{Z}, \mathbf{\Theta}\}$, and it follows that:

$$\mathbf{H}_k^{-1} = \mathrm{Diag}\left\{\mathbf{Z}^{-1},\ \mathbf{\Theta}^{-1}\right\}.$$

Since $\mathbf{Z}$ is diagonal, we have $\mathbf{Z}^{-1} = \mathrm{Diag}\{1/\frac{\partial g_\mu^2}{\partial z_n^2}, \forall n\}$, which clearly can be computed at each bus distributedly. Thus, the distributed computation of $\mathbf{H}_k^{-1}$ is reduced to computing the block $\mathbf{\Theta}^{-1}$ in a distributed fashion. However, this remains a challenging task since $\mathbf{\Theta}$ is dense and computing $\mathbf{\Theta}^{-1}$ require global information. One possible approach to handle this challenge is to borrow the matrix-splitting idea from [4–6] to compute $\mathbf{\Theta}^{-1}$ iteratively. However, the main drawbacks of this approach are: (i) The matrix-splitting scheme itself is a first-order method [29] and converges rather slowly (see numerical results in Section 5). This would defeat the whole purpose of our second-order algorithmic framework. (ii) The obtained solution under matrix-splitting is an approximation and only converges to the precise value *asymptotically.* This issue entails a *time-scale separation assumption* and it is impractical for implementations in power systems.

To overcome the limitations of matrix-splitting, we propose a *new* iterative approach based on the Sherman-Morrison-Woodbury (SMW) inversion lemma [2] stated as follows:

**Lemma 2** (SMW matrix inversion)**.** *For any invertible matrix* $\mathbf{P}$ *and vectors* $\mathbf{u}, \mathbf{v}$ *of conformable dimension, if* $1 + \mathbf{v}^T\mathbf{P}^{-1}\mathbf{u} \neq 0$*, then* $(\mathbf{P} + \mathbf{u}\mathbf{v}^T)^{-1}$ *can be computed as:*

$$(\mathbf{P} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{P}^{-1} - \frac{\mathbf{P}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{P}^{-1}}{1 + \mathbf{v}^T\mathbf{P}^{-1}\mathbf{u}}. \tag{18}$$

Note that if $\mathbf{P}^{-1}$ is known and the target matrix can be written as $\mathbf{P}$ coupled with a rank-1 update, then the formula in (18) provides a numerically cheap way to compute the result by a rank-1 correction based on $\mathbf{P}^{-1}$.

Now, upon a closer examination, we note that $\boldsymbol{\Theta}$ is not an arbitrary dense matrix but actually a *weighted Laplacian matrix* [28] of the underlying power grid network topology. To see this, we let $\gamma_l \triangleq \frac{2[\beta_l^2 + (\theta_{\mathrm{Tx}(l)} - \theta_{\mathrm{Rx}(l)})^2]}{[\beta_l^2 - (\theta_{\mathrm{Tx}(l)} - \theta_{\mathrm{Rx}(l)})^2]^2}$. Then, it can be verified that $\boldsymbol{\Theta}$ can be written as follows:

$$\boldsymbol{\Theta} = [-\mathbf{e}_N^{(1)}\ \mathbf{A}] \times \mathrm{Diag}\left\{\gamma_l,\, l = 0, \ldots, L\right\} \times [-\mathbf{e}_N^{(1)}\ \mathbf{A}]^T. \tag{19}$$

which shows that $\boldsymbol{\Theta}$ is a Laplacian matrix weighted by the $\gamma_l$-variables. Moreover, by expanding and collecting elements in (19) with respect to the rooted spanning tree, we can further decompose $\boldsymbol{\Theta}$ as follows:

$$\boldsymbol{\Theta} = \underbrace{\widetilde{\mathbf{A}}\mathrm{Diag}\left\{\gamma_l,\, l \in \mathcal{L}_0'\right\}\widetilde{\mathbf{A}}^T}_{\text{rooted spanning tree } \mathcal{T}} + \sum_{l \notin \mathcal{L}_0'} \gamma_l (\mathbf{A})_l [(\mathbf{A})_l]^T. \tag{20}$$

Now, it is important to recognize that (20) can be viewed as a weighted Laplacian matrix (weighted by $\gamma_l$, $l \in \mathcal{L}_0'$) of the rooted spanning tree $\mathcal{T}$ plus $L - (N-1)$ rank-1 updates. Thus, $\boldsymbol{\Theta}^{-1}$ can be obtained by performing SMW-corrections $L - (N-1)$ times on the inverse of the first term in (20). This motivates us to propose the first iterative SMW scheme to compute $\boldsymbol{\Theta}^{-1}$ as follows.

First, to handle the two parts of (20), we let $\sigma(k)$, $k = 0, \ldots, L$ denote a line ordering (fixing $\sigma(0) = 0$), where $\sigma(k) = l$ represents that the $k$-th position corresponding to line index $l$ in the electric power grid. The first $N$ positions in $\sigma(\cdot)$ correspond to the first part in (20), i.e., we let $\sigma(k) \in \mathcal{L}_0'$ for $k = 0, \ldots, N-1$. In addition, we let $\sigma(k)$ be such that $\widetilde{\mathbf{A}}$ is upper triangular if its columns are arranged following $\sigma(k)$. As an example, for the $\widetilde{\mathbf{A}}$ of the network in Figure 1(b), we have $\sigma(0) = 0$, $\sigma(1) = 2$, $\sigma(2) = 5$, $\sigma(3) = 1$, $\sigma(4) = 4$. Note that, once $\mathcal{T}$ (and hence $\widetilde{\mathbf{A}}$) is given, the first $N$ positions in $\sigma(\cdot)$ is *uniquely* specified because $\widetilde{\mathbf{A}}$ corresponds to a unique basis of the original matrix $\mathbf{A}$ [21]. Thus, the first $N$ positions in $\sigma(\cdot)$ can be determined by each bus on its own since they know $\widetilde{\mathbf{A}}$. The remaining $L - N + 1$ positions in $\sigma(\cdot)$ correspond to the second part in (20) and can be simply based on the row-ordering of $\widetilde{\mathbf{A}}$. For example, for the network in Figure 1, the last $6-(5-1)=2$ positions are for the non-spanning-tree lines $l_3$ and $l_6$. Note that $l_3$ starts from $n_4$ and $l_6$ starts from $n_5$. In $\widetilde{\mathbf{A}}$ of this example, the row for $n_4$ is above the row of $n_5$, thus we place $l_3$ before $l_6$, i.e., $\sigma(5)=3$ and $\sigma(6)=6$. Note that since each bus knows $\widetilde{\mathbf{A}}$, the last $L-N+1$ positions of $\sigma(\cdot)$ can also be assigned by each bus on its own. With $\sigma(\cdot)$, (20) can be rewritten as:

$$\begin{aligned}\mathbf{\Theta} = {} & \widetilde{\mathbf{A}}\mathrm{Diag}\left\{\gamma_{\sigma(k)},\, k=0,\ldots,N-1\right\}\widetilde{\mathbf{A}}^T \\ & + \sum_{i=1}^{L-N+1} \gamma_{\sigma(N-1+i)}(\mathbf{A})_{\sigma(N-1+i)}[(\mathbf{A})_{\sigma(N-1+i)}]^T. \end{aligned} \tag{21}$$

Now, we let

$$\mathbf{\Theta}_{(0)}^{-1} \triangleq \left[\widetilde{\mathbf{A}}\mathrm{Diag}\left\{\gamma_{\sigma(k)},\, k=0,\ldots,N-1\right\}\widetilde{\mathbf{A}}^T\right]^{-1}, \tag{22}$$

be the inverse of the first term in (21). Then, for $i=1,\ldots,L-N+1$, we can iteratively update $\mathbf{\Theta}_{(i)}^{-1}$ using the SMW inversion lemma as follows:

$$\mathbf{\Theta}_{(i)}^{-1} = \mathbf{\Theta}_{(i-1)}^{-1} - \frac{\gamma_{\sigma(N-1+i)}\mathbf{\Theta}_{(i-1)}^{-1}(\mathbf{A})_{\sigma(N-1+i)}(\mathbf{A})_{\sigma(N-1+i)}^T\mathbf{\Theta}_{(i-1)}^{-1}}{1+\gamma_{\sigma(N-1+i)}(\mathbf{A})_{\sigma(N-1+i)}^T\mathbf{\Theta}_{(i-1)}^{-1}(\mathbf{A})_{\sigma(N-1+i)}}. \tag{23}$$

Then, after $(L-N+1)$ iterations, we have $\mathbf{\Theta}^{-1} = \mathbf{\Theta}_{(L-N+1)}^{-1}$. Two remarks on the SMW scheme in (23) are in order: (i) From the $\sigma(\cdot)$-ordering definition, the last $L-(N-1)$ positions in $\sigma(\cdot)$ correspond to $l \notin \mathcal{T}$. However, the SMW scheme in (23) can still be performed on $\mathcal{T}$ bus-by-bus sequentially since $\mathcal{T}$ is a spanning tree and each bus knows exactly which lines incident on it are not in $\mathcal{T}$ and then handle those lines. (ii) Since $\mathbf{\Theta}_{(0)}^{-1}$ (cf. (22)) is symmetric and each SMW-update is symmetric (cf. (23)), $\mathbf{\Theta}_{(i)}^{-1}$ are symmetric, $\forall i$. Also, in each update in (23), since $(\mathbf{A})_{\sigma(N-1+i)}$ has exactly *two* non-zero elements 1 and $-1$, whose positions correspond to the two end nodes of line $\sigma(N-1+i)$, it will only pick up two rows in $\mathbf{\Theta}_{(i-1)}^{-1}$ corresponding to the two end nodes of line $\sigma(N-1+i)$ (see Appendix A for an illustrative example). As a result, we can let each bus $n$ store *only* the $n$-th row of $\mathbf{\Theta}_{(i)}^{-1}$ (columns are not needed since $\mathbf{\Theta}_{(i)}^{-1}$ is symmetric) and then the SMW scheme in (23) can be carried out as shown in Algorithm 1.

We can see from Algorithm 1 that the information exchange scale of the SMW scheme is dependent on the network diameter of $\mathcal{T}$. Thus, in rooted spanning tree generation, it is more advantageous to choose a $\mathcal{T}$ with small network diameter. We note that, although the information exchange scale of the SMW scheme is larger than matrix-splitting (1-hop [4]), the efficiency of the SMW scheme far outweighs matrix-splitting (see numerical results in Section 5). Thus, in the end, the amount of information exchanges under our SMW scheme is still much less than that of the matrix-splitting scheme.

**Algorithm 1** A bus-based sequential distributed scheme for computing $\boldsymbol{\Theta}^{-1}_{(i)}$.

```
Initialization:
  1. Given line ordering $\sigma(\cdot)$ where the last $(L-N+1)$ positions correspond to $l \notin \mathcal{T}$. Given that
     the $n$-th row of $\boldsymbol{\Theta}^{-1}_{(0)}$ is available at bus $n$, $\forall n$. Let $\gamma_l$ be available at Tx($l$), $\forall l$. Let $i = 1$.
Main Iteration:
  2. In the $i$-th iteration, at the transmitting bus of line $\sigma(N-1+i)$, use $\gamma_{\sigma(N-1+i)}$, the Tx($\sigma(N-1+i)$)-
     th row of $\boldsymbol{\Theta}^{-1}_{(i-1)}$ (both locally available) and the Rx($\sigma(N-1+i)$)-th row of $\boldsymbol{\Theta}^{-1}_{(i-1)}$ (available
     one-hop away) to compute $\boldsymbol{\Theta}^{-1}_{(i)}$ using (23).
  3. Send the $n$-th row of $\boldsymbol{\Theta}^{-1}_{(i)}$ to bus $n$ for each $n$. Send the complete $\boldsymbol{\Theta}^{-1}_{(i)}$ to the transmitting bus
     of $\sigma(N-i)$. Let $i \leftarrow i+1$. If $i = L-N$, stop; otherwise, repeat Step 2.
```

For the SMW scheme (23) to work, it remains to develop the distributed computations for $\boldsymbol{\Theta}^{-1}_{(0)}$ in (22). Note that

$$\boldsymbol{\Theta}^{-1}_{(0)} = (\widetilde{\mathbf{A}}^{-1})^T \mathrm{Diag}\left\{\frac{1}{\gamma_{\sigma(k)}},\, k = 0, \ldots, N-1\right\} \widetilde{\mathbf{A}}^{-1} \tag{24}$$

is a symmetric matrix. Due to the upper triangular structure of $\widetilde{\mathbf{A}}$ by Lemma 1, by expanding (24), we have

$$\left(\boldsymbol{\Theta}^{-1}_{(0)}\right)_{nn'} = \left(\boldsymbol{\Theta}^{-1}_{(0)}\right)_{n'n} = \sum_{k=1}^{n} \gamma_{\sigma(k-1)} (\widetilde{\mathbf{A}}^{-1})_{kn} (\widetilde{\mathbf{A}}^{-1})_{kn'}, \tag{25}$$

where $n \leq n'$. Using the structure in (25) and the properties of $\widetilde{\mathbf{A}}^{-1}$ in Lemma 1, we are able to develop a distributed scheme to compute $\boldsymbol{\Theta}^{-1}_{(0)}$ as follows: First, it is clear from the summation in (25) that bus $n$ only needs $\gamma_{\sigma(0)}, \ldots, \gamma_{\sigma(n-1)}$ in order to compute the $n$-th row of $\boldsymbol{\Theta}^{-1}_{(0)}$. Such information can be piggybacked starting from the artificial line $\sigma(0) = 0$ as follows: line $\sigma(0)$ sends $(\gamma_{\sigma(0)})$ to line $\sigma(1)$, line $\sigma(1)$ sends $(\gamma_{\sigma(0)}, \gamma_{\sigma(1)})$ to line $\sigma(2)$, so on and so forth. Since there are $N$ lines in $\mathcal{T}$ (including the artificial line), at most $N-1$ such $\gamma$-variables need to be piggybacked in the end. Also from Lemma 1, we have that $\widetilde{\mathbf{A}}^{-1}$ only has ±1 and 0 elements. The above observations give rise to a *bus-based sequential* distributed scheme to compute $\boldsymbol{\Theta}^{-1}_{(0)}$ as illustrated in Algoirthm 2. As we can see in Algorithm 2, in the $n$-th iteration, bus $n$ only needs to compute entries $(\boldsymbol{\Theta}^{-1}_{(0)})_{nn'}$ in the $n$-th row of $\boldsymbol{\Theta}^{-1}_{(0)}$, with $n \leq n'$. The remaining $n-1$ have been computed by buses $1, \ldots, n-1$ and relayed to bus $n$. In particular, we can see from Step 2 that to compute $(\boldsymbol{\Theta}^{-1}_{(0)})_{nn'}$, bus $n$ just needs to compare signs of the elements (since the elements are ±1 and 0 only) in the $n$-th and $n'$-th columns of $\widetilde{\mathbf{A}}^{-1}$, and then decide whether to add or subtract $\gamma_{\sigma(k-1)}$. Therefore, Algorithm 2 is highly efficient and finishes computing $\boldsymbol{\Theta}^{-1}_{(0)}$ in $N$ iterations.

Finally, combining Algorithm 1 and Algorithm 2, we get the *precise* solution of $\boldsymbol{\Theta}^{-1}$ in *exactly* $L$

**Algorithm 2** A bus-based sequential distributed scheme for computing $\mathbf{\Theta}_{(0)}^{-1}$.

**Initialization:**

1. Given a line ordering $\sigma(\cdot)$ such that $\widetilde{\mathbf{A}}$ with its columns arranged in $\sigma(\cdot)$ is upper triangular. Let $n = 1$.

**Main Iteration:**

2. In the $n$-th iteration, bus $n$ does: for $k = 1, \ldots, n$ and for all $n' \geq n$, if $(\widetilde{\mathbf{A}}^{-1})_{kn}, (\widetilde{\mathbf{A}}^{-1})_{kn'} \neq 0$ and $(\widetilde{\mathbf{A}}^{-1})_{kn}$ and $(\widetilde{\mathbf{A}}^{-1})_{kn'}$ are of the same sign, then
$$(\mathbf{\Theta}_{(0)}^{-1})_{nn'} = (\mathbf{\Theta}_{(0)}^{-1})_{nn'} + \gamma_{\sigma(k-1)}.$$
Otherwise, if $(\widetilde{\mathbf{A}}^{-1})_{kn}, (\widetilde{\mathbf{A}}^{-1})_{kn'} \neq 0$ and $(\widetilde{\mathbf{A}}^{-1})_{kn}$ and $(\widetilde{\mathbf{A}}^{-1})_{kn'}$ are of the opposite sign, then
$$(\mathbf{\Theta}_{(0)}^{-1})_{nn'} = (\mathbf{\Theta}_{(0)}^{-1})_{nn'} - \gamma_{\sigma(k-1)}.$$
3. If $n = N$ stop. Otherwise, send $(\mathbf{\Theta}_{(0)}^{-1})_{nn'}$, $n \leq n'$, and $\gamma_{\sigma(k-1)}$, $k = 1, \ldots, n$, to bus $n+1$; let $n \leftarrow n+1$ and repeat Step 2.

iterations. With $\mathbf{\Theta}^{-1}$, the primal Newton directions of Problem (13) can be computed distributedly as shown in the following theorem:

**Theorem 3.** *Given dual variables $\mathbf{w}^k$, the primal Newton directions $\Delta z_n^k$ and $\Delta\theta_n^k$ can be computed using local information at each bus $n$. More specifically, $\Delta z_n^k$ and $\Delta\theta_n^k$ can be computed as follows: for $n = 1, \ldots, N$,*

$$\Delta z_n^k = \frac{z_n(K_n - z_n)}{z_n^2 + (K_n - z_n)^2}(K_n - 2z_n + z_n(K_n - z_n)\times \\ (w_n - \mu C_n'(z_n))), \tag{26}$$

$$\Delta\theta_n^k = \sum_{n'=1}^{N} (\mathbf{\Theta}^{-1})_{nn'} \Big( \sum_{l \in \Psi(n)} \gamma_l + \sum_{l \in \Psi(n) \cup \mathcal{L}_0'} b_l (w_{\mathcal{E}(l)\backslash n}^k - w_n^k) \Big). \tag{27}$$

*Proof.* It can be verified that the results in (26) and (27) follow from expanding (16) and exploiting the structure of $\mathbf{M}$ to simplify the expansion of (16). Also, we can see that (26) and (27) only involve local information $z_n$, $\gamma_l$, $l \in \Psi(n)$, and $w_n$ and $w_{\mathcal{E}(l)\backslash n}$ that are either available at bus $n$ or at one-hop neighbor of bus $n$. This completes the proof. □

**Remark 1.** *In (27), we can think of the difference of dual variables $(w_{\mathcal{E}(l)\backslash n}^k - w_n^k)$ as being analogous to the "nodal potential differential" in the network simplex algorithm [21], which is in essence a first-order gradient-based steepest descent method for linear max-flow problems. However, unlike first-order methods, our decision to change $\theta_n^k$ considers not just the "potential differential" of one line, but all* *lines incident upon bus n. This observation shows the key connection and difference between first- and second-order methods.*

**b) Distributed computation of dual variables.** Recall that, given a primal solution $\mathbf{y}^k$ at the $k$-th iteration, the dual variables $\mathbf{w}^k$ can be computed using (17). However, as mentioned earlier, it remains unclear how to solve for $\mathbf{w}^k$ in a distributed fashion because $(\mathbf{M}\mathbf{H}_k\mathbf{M}^T)^{-1}$ is dense and computing $(\mathbf{M}\mathbf{H}_k\mathbf{M}^T)^{-1}$ requires global information. In what follows, we will first examine the structure of $\mathbf{M}\mathbf{H}_k\mathbf{M}^T$. From the definitions of $\mathbf{M}$ and $\mathbf{H}_k^{-1}$, $\mathbf{M}\mathbf{H}_k\mathbf{M}^T$ can be written in blockwise form as:

$$\mathbf{M}\mathbf{H}_k\mathbf{M}^T=\begin{bmatrix} (\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^T)\mathbf{\Theta}^{-1}(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^T)+\mathbf{Z}^{-1} & \mathbf{Z}^{-1}\mathbf{1} \\ \mathbf{1}^T\mathbf{Z}^{-1} & \mathbf{1}^T\mathbf{Z}^{-1}\mathbf{1} \end{bmatrix}. \tag{28}$$

Using the blockwise matrix inversion theorem [30], we have that:

$$(\mathbf{M}\mathbf{H}_k\mathbf{M}^T)^{-1}=\begin{bmatrix} \mathbf{S}^{-1} & \mathbf{Q}_1 \\ \mathbf{Q}_1^T & \mathbf{Q}_2 \end{bmatrix}, \tag{29}$$

where $\mathbf{Q}_1 = -\mathbf{S}^{-1}\mathbf{Z}^{-1}\mathbf{1}(\mathbf{1}^T\mathbf{Z}^{-1}\mathbf{1})^{-1}$ and
$\mathbf{Q}_2=(\mathbf{1}^T\mathbf{Z}^{-1}\mathbf{1})^{-1}+(\mathbf{1}^T\mathbf{Z}^{-1}\mathbf{1})^{-1}\mathbf{1}^T\mathbf{Z}^{-1}\mathbf{S}^{-1}\mathbf{Z}^{-1}\mathbf{1}(\mathbf{1}^T\mathbf{Z}^{-1}\mathbf{1})^{-1}$.

In (29), $\mathbf{S}$ is the Schur complement [30] of $\mathbf{M}\mathbf{H}_k\mathbf{M}^T$ and can be further written as:

$$\mathbf{S}=(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^T)\mathbf{\Theta}^{-1}(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^T)+\mathbf{Z}^{-1}-\mathbf{Z}^{-1}\mathbf{1}(\mathbf{1}^T\mathbf{Z}^{-1}\mathbf{1})^{-1}\mathbf{1}\mathbf{Z}^{-1}. \tag{30}$$

In (29), since $\mathbf{Z}$ is diagonal and due to the simple structure of $\mathbf{1}$, the computations of $\mathbf{Q}_1$ and $\mathbf{Q}_2$ can be decentralized to each node once $\mathbf{S}$ is known. Thus, the distributed computation of $(\mathbf{M}\mathbf{H}_k\mathbf{M}^T)^{-1}$ boils down to decentralizing the inverse of the Schur complement $\mathbf{S}$.

Note that since the quantity $(\mathbf{1}^T\mathbf{Z}^{-1}\mathbf{1})^{-1}$ is a scalar, the last term in (30) can be written as a rank-1 update as follows: $(\mathbf{1}^T\mathbf{Z}^{-1}\mathbf{1})^{-1}(\mathbf{Z}^{-1}\mathbf{1})(\mathbf{Z}^{-1}\mathbf{1})^T$ . Then, we have the following decomposition of $\mathbf{S}$ by noting $\mathbf{Z}$ is diagonal:

$$\mathbf{S} = (\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^T)\mathbf{\Theta}^{-1}(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^T) + \sum_{n=1}^{N}(\mathbf{Z}^{-1})_{nn}\mathbf{e}_N^{(n)}(\mathbf{e}_N^{(n)})^T - (\mathbf{1}^T\mathbf{Z}^{-1}\mathbf{1})^{-1}(\mathbf{Z}^{-1}\mathbf{1})(\mathbf{Z}^{-1}\mathbf{1})^T. \tag{31}$$

Now, it is important to recognize from (31) that $\mathbf{S}$ can be viewed as $(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^T)\mathbf{\Theta}^{-1}(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^T)$ coupled with $N+1$ rank-1 updates. Thus, we can use a *second layer* of SMW scheme to compute $\mathbf{S}^{-1}$ as follows. First, we let

$$\mathbf{S}_{(0)}^{-1} = \left[(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^T)\mathbf{\Theta}^{-1}(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^T)\right]^{-1}, \tag{32}$$

which can be further written as:

$$\begin{aligned}&\Big[(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^T)\mathbf{\Theta}^{-1}(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^T)\Big]^{-1} \overset{(a)}{=} (\widetilde{\mathbf{A}}^{-1})^T\widetilde{\mathbf{B}}^{-1}\widetilde{\mathbf{A}}^{-1}\times \\ &\Big[\widetilde{\mathbf{A}}\mathrm{Diag}\left\{\gamma_l, l\in\mathcal{L}_0'\right\}\widetilde{\mathbf{A}}^T + \sum_{l\in\mathcal{L}_0'}\gamma_l(\mathbf{A})_l(\mathbf{A})_l^T\Big](\widetilde{\mathbf{A}}^T)^{-1}\widetilde{\mathbf{B}}^{-1}\widetilde{\mathbf{A}}^{-1} \\ &= (\widetilde{\mathbf{A}}^{-1})^T\mathrm{Diag}\left\{\frac{\gamma_l}{b_l^2}, l\in\mathcal{L}_0'\right\}\widetilde{\mathbf{A}}^{-1} + \\ &\sum_{l\notin\mathcal{L}_0'}\gamma_l(\widetilde{\mathbf{A}}^{-1})^T\widetilde{\mathbf{B}}^{-1}\widetilde{\mathbf{A}}^{-1}(\mathbf{A})_l(\mathbf{A})_l^T(\widetilde{\mathbf{A}}^{-1})^T\widetilde{\mathbf{B}}^{-1}\widetilde{\mathbf{A}}^{-1},\end{aligned} \tag{33}$$

where the equality $(a)$ follows from (20). Note that every bus has the knowledge of $\widetilde{\mathbf{A}}$ (and hence $\widetilde{\mathbf{A}}^{-1}$) and $\widetilde{\mathbf{B}}$ after the rooted spanning tree identification. Again, note that the column vector $(\mathbf{A})_l$ in the second term in (33) has exactly two non-zero elements 1 and $-1$ that correspond to buses $\mathrm{Tx}(l)$ and $\mathrm{Rx}(l)$ respectively. Further, due to the *same* decomposition structure between (33) and (20), we have that (33) can be computed along with $\mathbf{\Theta}^{-1}$ in (22) and (23), thus saving a significant amount of extra message passings.

Next, given $\mathbf{S}_{(0)}^{-1}$ and following the SMW matrix inversion lemma, we have for $n = 1, \ldots, N$,

$$\mathbf{S}_{(n)}^{-1} = \mathbf{S}_{(n-1)}^{-1} - \frac{(\mathbf{Z}^{-1})_{nn}\mathbf{S}_{(n-1)}^{-1}\mathbf{e}_N^{(n)}(\mathbf{e}_N^{(n)})^T\mathbf{S}_{(n-1)}^{-1}}{1+(\mathbf{Z}^{-1})_{nn}\mathbf{e}_n^T\mathbf{S}_{(n-1)}^{-1}\mathbf{e}_n}. \tag{34}$$

After $N$ iterations, we perform the final rank-1 correction to incorporate the last term in (31) to obtain:

$$\mathbf{S}^{-1} = \mathbf{S}_{(N)}^{-1} + \frac{(\mathbf{1}^T\mathbf{Z}^{-1}\mathbf{1})\mathbf{Z}^{-1}\mathbf{1}\mathbf{1}^T\mathbf{Z}^{-1}}{1-(\mathbf{1}^T\mathbf{Z}^{-1}\mathbf{1})\mathbf{1}^T\mathbf{Z}^{-1}\mathbf{1}}. \tag{35}$$

With the computation scheme in (29), (34), and (35), we have the *precise* result of $(\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T)^{-1}$ in *exactly* $N+1$ iterations. Finally, combining all related discussions, we can derive the following result:

**Theorem 4.** *Given primal variables $\mathbf{y}^k$, the dual variables $\mathbf{w}^k$ can be computed at each bus $n$ as follows:*

$$w_n^k = \sum_{n'=1}^{N}\Big[((\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T)^{-1})_{nn'}\Big(-q_{n'} - \sum_{l\in\Psi(n')\cap\mathcal{L}_0'} b_l(\eta_{n'} - \eta_{\mathcal{E}(l)\backslash n'})\Big)\Big], \quad n = 1, \ldots, N, \tag{36}$$

*where we define $q_n \triangleq \frac{2z_n+p_n-\mu z_n(z_n+p_n)C_n'(z_n)}{z_n^2+(z_n+p_n)^2+\mu z_n^2(z_n+p_n)^2C_n''(z_n)}z_n(z_n+p_n)$ and $\eta_n \triangleq \sum_{n\in\Psi(n)}\sum_{n'=1}^{N}\gamma_n(\mathbf{\Theta}^{-1})_{nn'}$.*

*Proof.* The result in (36) follows from the entry-wise expansion of (17) and exploiting the structure of $\mathbf{M}$ to simplify the term $-\mathbf{M}\mathbf{H}_k^{-1}\nabla g_\mu(\mathbf{y}^k)$. We omit the detailed derivations here due to space

**Algorithm 3** A double SMW distributed algorithm for computing primal Newton directions and dual variables.

```
Initialization:
  1. Identify a rooted spanning tree, for which the NAIM is denoted as $\widetilde{\mathbf{A}}$. Given a line ordering $\sigma(\cdot)$
     such that $\widetilde{\mathbf{A}}$ with its columns arranged in $\sigma(\cdot)$ is upper triangular.
Main Iteration:
  2. Primal Newton directions: Compute $\mathbf{\Theta}_{(0)}^{-1}$ by using Algorithm 2. For $i = 1, \ldots, L-N+1$, compute
     $\mathbf{\Theta}^{-1}$ using the SMW scheme in (23) by Algorithm 1. Then, update $\Delta z_n^k$ and $\Delta\theta_n^k$ using (26) and
     (27) in Theorem 3. Meanwhile, compute the initial Schur complement value $\mathbf{S}_{(0)}^{-1}$ using (33) for
     later use in Step 3.
  3. Dual variables updates: For $n = 1, \ldots, N$, compute $(\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T)^{-1}$ using the SMW scheme in
     (29), (34), and (35) in a bus-by-bus sequential fashion. Then, update $w_n^k$ using (36) in Theorem 4.
```

limitation. To see how (36) can be carried out in a distributed fashion, note that since the computation scheme (34) for $(\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T)^{-1}$ can be done in a node-by-node fashion on the rooted spanning tree $\mathcal{T}$, the $q_n$- and $\eta_n$-information (both involves local information only) can be piggybacked along with $\mathbf{S}_{(n)}^{-1}$ in (34) at the same time. Hence, by the time the computation of $\mathbf{S}^{-1}$ is complete, the required $q_n$- and $\eta_n$-information is also available at each bus, which further suggests that (36) can be computed at each bus. □

To conclude the discussions of distributed algorithm design, we summarize the computation schemes of primal Newton directions and dual variables in Algorithm 3. In Algorithm 3, after initialization in Step 1, Steps 2 is the first layer of SMW scheme for computing primal Newton directions $\Delta\mathbf{y}^k$ in (16); while Step 3 is the second layer of SMW scheme for dual updates $\mathbf{w}^k$ in (17).

### 4.3 Algorithm Initialization

So far, we have developed a distributed double SMW scheme to compute the primal Newton directions and dual variables. Another design element remained to be addressed is to identify a starting point. In this paper, we propose a maximum scaling factor scheme to bootstrap our second-order distributed scheme.

Toward this end, we let $\widehat{\boldsymbol{\theta}} = [\widehat{\theta}_1, \ldots, \widehat{\theta}_N]^T \in \mathbb{R}^N$ be the phase angles before load shedding. Since $\widehat{\boldsymbol{\theta}}$ may cause power overflow, a set of initial operating angles *within* the new feasible domain is needed for our interior-point based load-shedding scheme. Thanks to the linearity of the DC model, a simple way of initialization is to shrink all $\widehat{\theta}$-variables by the same scaling factor $0 < \alpha < 1$, i.e., $\mathbf{f} = \mathbf{B}\mathbf{A}^T(\alpha\widehat{\boldsymbol{\theta}})$. Clearly, there always exists a small enough $\alpha$ such that $\mathbf{f}$ will not exceed any line capacity (i.e., in the interior of the new feasible domain). However, for a small blackout impact at the initial point, we hope to find the maximum $\alpha$ while not exceeding bounding limits. This can be formulated as:

$$
\begin{aligned}
&\text{Maximize} && \alpha && \\
&\text{subject to} && \mathbf{f} = \mathbf{B}\mathbf{A}^T\widehat{\boldsymbol{\theta}}\alpha, && && \text{(37a)} \\
& && |f_l| \leq \min\{f_l^{\max}, b_l\beta\}, && \forall l, && \text{(37b)} \\
& && (\mathbf{A}\mathbf{f})_n \leq B_n^{\max}, && \forall n \in \mathcal{K}. && \text{(37c)}
\end{aligned}
$$

Rewriting (37a) in scalar form and substituting it into (37b) and (37c), we have $\alpha b_l|\widehat{\theta}_{\mathrm{Tx}(l)} - \widehat{\theta}_{\mathrm{Rx}(l)}| \leq \max\{f_l^{\max}, b_l\beta\}$, $\forall l$, and $\alpha \sum_{\Psi(n)\cap\mathcal{L}_0'} b_l(\widehat{\theta}_n - \widehat{\theta}_{\mathcal{E}(l)\backslash n}) \leq B_n^{\max}$. It then follows that the optimal scaling factor $\alpha^*$ can be computed as:

$$
\alpha^* = \min_{\substack{\forall l \\ n\in\mathcal{K}}} \left\{ \frac{\min\{f_l^{\max}, b_l\beta\}}{b_l|\widehat{\theta}_{\mathrm{Tx}(l)} - \widehat{\theta}_{\mathrm{Rx}(l)}|}, \frac{B_n^{\max}}{\sum_{\Psi(n)\cap\mathcal{L}_0'} b_l(\widehat{\theta}_n - \widehat{\theta}_{\mathcal{E}(l)\backslash n})} \right\}. \tag{38}
$$

Note that $\alpha^*$ can be rewritten as: $\alpha^* = \min\{\min_{\forall n}\{\alpha_{n,1}^*\}, \min_{n\in\mathcal{K}}\{\alpha_{n,2}\}\}$, where

$$
\alpha_{n,1}^* \triangleq \min_{l\in\mathcal{O}(n)} \left\{ \frac{\min\{f_l^{\max}, b_l\beta\}}{b_l|\widehat{\theta}_{\mathrm{Tx}(l)} - \widehat{\theta}_{\mathrm{Rx}(l)}|} \right\}
$$

and $\alpha_{n,2} = \frac{B_n^{\max}}{\sum_{\Psi(n)\cap\mathcal{L}_0'} b_l(\widehat{\theta}_n - \widehat{\theta}_{\mathcal{E}(l)\backslash n})}$. This implies that each bus can first determine $\alpha_{n,1}^*$ and $\alpha_{n,2}$ locally, and then broadcasts $\alpha_{n,1}^*$ and $\alpha_{n,2}$ onto the rooted spanning tree $\mathcal{T}$. After hearing all $\alpha_{n,1}^*$- and $\alpha_{n,2}$-information, each bus can determine the optimal scaling factor $\alpha^*$.

### 4.4 Step-Size Control and Problem LSO

In this subsection, we consider step-size selection strategy and show how to use step-size control to obtain solutions to the original Problem LSO. First, we note that to solve the original Problem LSO using our proposed second-order distributed approach, we can replace the iteration index "$k$" in (16) and (17) by time-slot index "$[t]$" and with a statistical estimate of $\mathbf{p}$ based on time-average (or historical data if the demands are predictable). Then, we have

$$
\Delta\mathbf{y}_{[t]} = -\mathbf{H}_{[t]}^{-1}(\nabla g_\mu(\mathbf{y}_{[t]}) + \mathbf{M}^T\mathbf{w}_{[t]}), \tag{39}
$$

$$
\mathbf{w}_{[t]} = (\mathbf{M}\mathbf{H}_{[t]}^{-1}\mathbf{M}^T)^{-1}(-\mathbf{M}\mathbf{H}_{[t]}^{-1}\nabla g_\mu(\mathbf{y}_{[t]})), \tag{40}
$$

and the algorithm evolves with time as:

$$
\mathbf{y}_{[t+1]} = \mathbf{y}_{[t]} + s_{[t]}\Delta\mathbf{y}_{[t]}. \tag{41}
$$

Clearly, as the time average approaches the true value of $\mathbf{p}$ (recall that $\mathbf{p}_{[t]}$ is ergodic), the solution obtained under (39), (40), and (41) with some appropriate step-size selection $s_{[t]}$ also approaches the optimal solution. In the interior-point methods literature [27], there are two standard step-size selection rules. The first one is to guarantee that the iterates stay inside the interior of the feasible domain. In the context of our problem, this means choosing $s_{[t]}$ as:

$$\sup \left\{ s \in (0,1] \;\middle|\; \begin{array}{l} z_n[t] + s\Delta z_n[t] < K_n, \, \forall n, \\ |\mathbf{A}^T(\boldsymbol{\theta}_{[t]} + s\Delta\boldsymbol{\theta}_{[t]})| < \boldsymbol{\beta}. \end{array} \right\}, \tag{42}$$

where $\boldsymbol{\beta} \triangleq [\beta_1, \ldots, \beta_L]^T \in \mathbb{R}^L$. The second rule is to guarantee a "decreasing merit function" [27]. More specifically, let $r_\mu(\mathbf{y}_{[t]}, \mathbf{w}_{[t]}) \triangleq \nabla g_\mu(\mathbf{y}_{[t]}) + \mathbf{M}^T\mathbf{w}_{[t]}$ denote the residual of the stationarity condition in the Karash-Kuhn-Tucker (KKT) system at the current iterate $(\mathbf{y}_{[t]}, \mathbf{w}_{[t]})$, which measures closeness between $(\mathbf{y}_{[t]}, \mathbf{w}_{[t]})$ and the optimal primal-dual pair $(\mathbf{y}^*, \mathbf{w}^*)$ that satisfies $\nabla g_\mu(\mathbf{y}^*) + \mathbf{M}^T\mathbf{w}^* = \mathbf{0}$. Then, we choose $s_{[t]}$ to enforce $\|r_\mu(\mathbf{y}_{[t+1]}, \mathbf{w}_{[t+1]})\| \leq \|r_\mu(\mathbf{y}_{[t]}, \mathbf{w}_{[t]})\|$. Under these two step-size rules, the convergence and the second-order convergence rate analysis follow from standard interior point methods [27].

However, we note that in practical implementations, the second step-size selection rule, i.e., the "decreasing merit function," is *expensive* to check and hard to decentralize due to a large number of gradient and dual evaluations in each time slot. Thus, in this paper, we relax the decreasing residual rule, while keeping the feasibility rule only (a basic requirement in an interior-point method). Note that, due to this relaxed step-size rule, the sequence $(\mathbf{y}_{[t]})_{t=0}^{\infty}$ is not guaranteed to converge. However, we can show that their long-term average does converge to a bounded region around the optimal solution $\mathbf{y}^*$ as indicated in Theorem 5, which is *exactly* the objective in Problem LSO.

**Theorem 5.** *Let $\mathbf{y}^*$ represent an optimal solution to Problem LSO. Let $\mu$ be a given central-path tracking parameter. Under the second-order scheme in (39), (40), and (41) with the step-size selection rule (42), there exists a constant $B < \infty$ such that $\limsup_{T\to\infty} |\frac{1}{T}\sum_{t=0}^{T-1} \mathbf{y}_{[t]} - \mathbf{y}^*| \leq B$.*

*Proof.* The main idea and key steps for proving Theorem 5 are as follows. First, we consider the following choice of Lyapunov function: $V(\mathbf{y}_{[t]}) \triangleq \frac{1}{2}\|\mathbf{y}_{[t]} - \mathbf{y}^*\|^2$, which can be interpreted as measuring the distance between the current iterate $\mathbf{y}_{[t]}$ to the optimal solution $\mathbf{y}^*$. Then, after some algebraic derivations and upper-bounding (see Appendix B), we obtain the following relationship:

$$\begin{aligned} \Delta V(y_{[t]}) &\triangleq V(\mathbf{y}_{[t+1]}) - V(\mathbf{y}_{[t]}) \\ &\leq -s^{\min}\|\mathbf{y}_{[t]} - \mathbf{y}^*\|^2 + B_1 + B_2, \end{aligned} \tag{43}$$

where $s^{\min} \triangleq \inf_t\{s_{[t]}\} > 0$ represents the lower bound of the step-sizes. Note that such a lower bound must exist since the iterates $\{\mathbf{y}_{[t]}\}_{t=1}^{\infty}$ always take $s_{[t]} = 1$ (i.e., a full Newton step) when they are far from the feasible domain boundary, and $s_{[t]}$ starts to decrease when $\{\mathbf{y}_{[t]}\}_{t=1}^{\infty}$ approach the boundary. However, due to the barrier terms in (12), there exists a point close to the boundary, beyond which the barrier terms dominate the original objective function and yield a (Hessian-deflected) gradient direction pointing inward away from the boundary. When this happens, a full Newton step will be taken again, implying that $s_{[t]}$ *cannot* decrease to zero asymptotically. In (43), $B_1$ and $B_2$ are some positive constants defined as follows:

$$B_1 \triangleq \frac{s^{\min}}{\lambda_{\min}(\mathbf{H})} \sup_t \{\|\mathbf{y}_{[t]} - \mathbf{y}^*\| \|\mathbf{M}^T(\mathbf{w}_{[t]} - \mathbf{w}^*)\|\}.$$

$$B_2 \triangleq \frac{1}{2\lambda^2_{\min}(\mathbf{H})} \sup_t \big(\|g_\mu(\mathbf{y}_{[t]}) - g_\mu(\mathbf{y}^*)\| \\ + \|\mathbf{M}^T(\mathbf{w}_{[t]} - \mathbf{w}^*)\|\big)^2.$$

Telescoping $T$ inequalities in (43) for $t = 0, \ldots, T-1$ yields:

$$V(\mathbf{y}_{[T]}) - V(\mathbf{y}_{[0]}) \leq -s^{\min} \sum_{t=0}^{T-1} \|\mathbf{y}_{[t]} - \mathbf{y}^*\|^2 + T(B_1 + B_2).$$

Dividing both sides by $Ts^{\min}$, rearranging terms, and taking $T$ to infinity, we have

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \|\mathbf{y}_{[t]} - \mathbf{y}^*\|^2 \leq B^2,$$

where we define $B^2 \triangleq \frac{B_1 + B_2}{s^{\min}}$. Then, the proof is complete because when $T$ is large, we have

$$\Big|\frac{1}{T} \sum_{t=0}^{T-1} (\mathbf{y}_{[t]} - \mathbf{y}^*)\Big| \overset{(a)}{\leq} \Big(\frac{1}{T} \sum_{t=0}^{T-1} \|\mathbf{y}_{[t]} - \mathbf{y}^*\|^2\Big)^{\frac{1}{2}} \leq B,$$

where $(a)$ follows from the triangular inequality and the basic relationship between $l_1$- and $l_2$-norms. We note that the most involved step in the proof lies in the one-slot drift analysis, where we repeatedly exploit the KKT stationary condition $\nabla g_\mu(\mathbf{y}^*) + \mathbf{M}^T \mathbf{w}^* = 0$. We relegate the full proof to Appendix B. □

**Remark 2.** *Theorems 5 shows that even with the relaxed step-size selection strategy (more suitable for implementation in practice), our proposed distributed second-order load shedding scheme can still achieve a near optimal performance in average sense. We note that this relaxed step-size strategy is one of the main novelties in this work and, to the best of our knowledge, the performance guarantee result under this relaxed step-size strategy has not been reported in the interior-point methods literature.*

# 5 Numerical Results

In this section, we present some numerical results to demonstrate the practicality and efficacy of our proposed distributed second-order load shedding algorithm. To illustrate the convergence speed of our algorithm, we use the IEEE 30-bus benchmark system [31] as illustrated in Figure 2. The IEEE 30-bus system is a portion of the American Electric Power System (AEP) in the midwest of US [31]. In this 30-bus benchmark system, the generator buses are 1, 2, 5, 8, 11, 13, and the data for this benchmark system (in per-unit with base power rating 1 MVA) are extracted from the software toolbox "MATPOWER" [24]. The demand at each remaining load buses is randomly generated with mean equal to 1 unit. Here, we suppose that this 30-bus system loses both its electric and communication connections from the centralized controller after a disaster strike. As a result, the 30-bus system has to perform distributed load shedding on its own in a decentralized fashion. The cost of load shedding at each load bus $n$ is measured by a positively weighted quadratic function $\omega_n z_n^2$ with weights $\omega_n$ randomly generated between 0 and 1.

For this 30-bus benchmark system, the convergence processes of our proposed distributed second-order load shedding algorithm with different choices of initial scaling factors $\alpha$ are illustrated in Figure 3. We can see that, in all cases, our distributed second-order method converges in less than 35 iterations, which demonstrates the powerful second-order convergence speed. From Figure 3, we can also observe that the choice of initial starting points (controlled by the scaling factor $\alpha$) can further affect the convergence speed of the algorithm. In this example, the maximum scaling factor turns out to be 0.62, which corresponds to the black curve in Figure 3. We can see that, with the maximum scaling factor, the algorithm converges in less than 25 time slots.

To see the effect of load shedding optimization, we use bus 6 as an example. After optimization, the load shedding amount is reduced by 0.26 unit (260 KW). To put this number into perspective, 260 KW is more than enough for the combined consumption of emergency communications department, fire department, and the traffic and transportation department in Cambridge, Massachusetts (170 KW in total [32]), all of which are critical in disaster recovery.

Next, we compare the convergence performance between our proposed double SMW scheme with the matrix-splitting scheme in [4]. For the IEEE 30-bus benchmark system (41 lines), the convergence processes of our double SMW and the matrix-splitting scheme are illustrated in Figure 4, where the top portion is for primal Newton directions and the bottom portion is is for dual variables. In Figure 4, the matrix-splitting algorithm is terminated when the errors with respect to the true values of $\Delta\mathbf{y}$ and $\mathbf{w}$ are less than $10^{-6}$. We can see from Figure 4 that, as expected, our double SMW scheme obtain the precise solution of $\Delta\mathbf{y}$ and $\mathbf{w}$ in 41 and 31 iterations, respectively. In contrast, for the 30-bus benchmark system, the matrix-splitting scheme takes nearly $10^4$ and more than $8 \times 10^4$ iterations to reach the $10^{-6}$ precision. In other words, the numbers of iterations required by the matrix-splitting scheme are three orders of magnitude larger than those of the double SMW scheme. This suggests that the matrix-splitting scheme is *impractical* for fast distributed load shedding in practice.

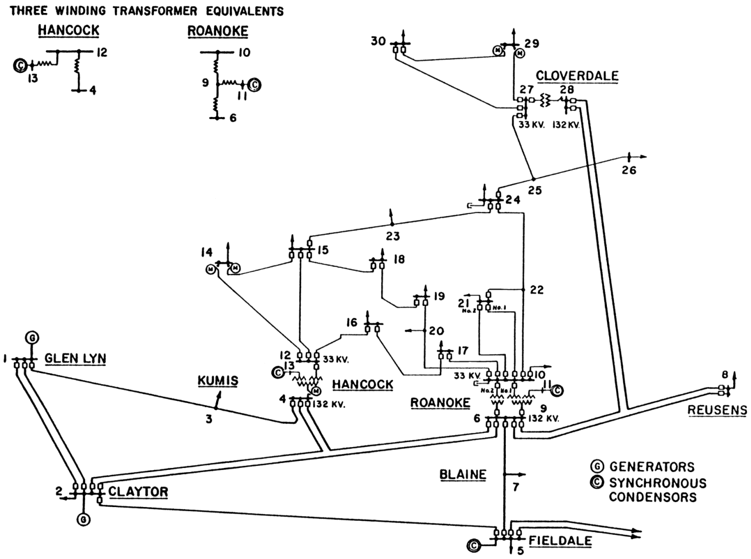

Figure 2: The network topology of the IEEE 30-bus benchmark system taken from [31].

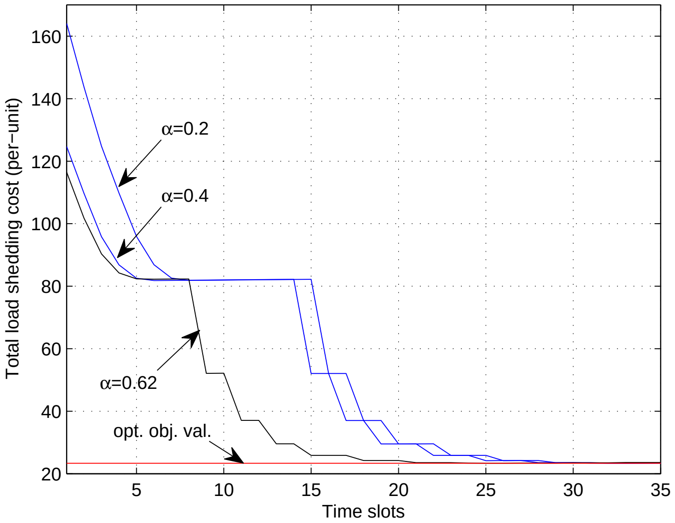

Figure 3: The convergence process of the proposed distributed second-order load shedding algorithm with different choices of initial scaling factors.

To compare the convergence performances between our distributed second-order scheme with the first-order gradient based search scheme, we randomly generated 50 60-bus network examples. The numbers of iterations for both algorithms are plotted in Figure 5. For these 60-bus test systems, the average numbers of iterations for our proposed distributed second-order method and the first-order subgradient method are 55.2 and 9351.2, respectively. This shows that our proposed algorithm converges more than two orders of magnitude faster. Such a fast convergence speed is not only

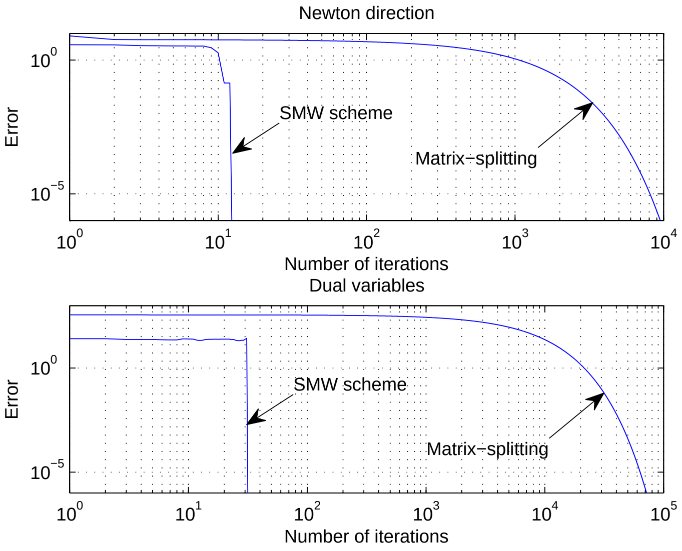

Figure 4: The convergence performance comparison between our proposed double SMW scheme and the matrix-splitting scheme in [4].

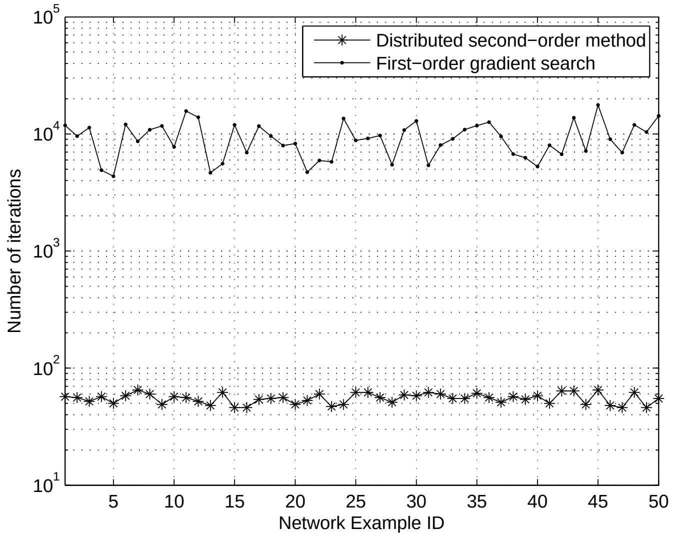

Figure 5: Convergence speed comparison between the proposed second-order method and the first-order gradient based method.

desirable but also necessary for post-disaster load shedding.

# 6 Conclusion

In this paper, we developed a new distributed second-order load shedding optimization algorithm for disaster recovery in electric power grids, We first proposed a rooted spanning tree based reformulation that reduces the problem size and enables our distributed second-order algorithm design. Then, based on the problem structure of the rooted spanning tree reformulation, we developed a double Sherman-Morrison-Woodbury (SMW) scheme that yields distributed computation solutions for primal Newton directions and dual variables. Further, we designed an efficient distributed scaling factor scheme to initialize our second-order load shedding scheme. Finally, we propose a relaxed step-size selection strategy, which is well-suited for distributed implementation in practice and still provides optimality performance guarantee. To verify the efficacy of our proposed algorithm, we conduct extensive numerical studies. Our results showed that the proposed converges more than two orders of magnitude faster than the first-order gradient based search methods. Collectively, our results serve as the first building block of a new second-order theoretical framework for distributed optimization in future smart grids.

# References

[1] M. Amin and P. F. Schewe, "Preventing blackouts: Building a smarter power grid," *Scientific American*, May 2008. [Online]. Available: http://www.scientificamerican.com/article.cfm?id=preventing-blackouts-power-grid

[2] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, 3rd ed. New York, NY: John Wiley & Sons Inc., 2006.

[3] A. Jadbabaie, A. Ozdaglar, and M. Zargham, "A distributed Newton method for network optimization," in *Proc. IEEE Conference on Decision and Control (CDC)*, Shanghai, China, Dec. 16-18, 2009.

[4] E. Wei, A. Ozdaglar, and A. Jadbabaie, "A distributed newton method for network utility maximization, I: Algorithm," *IEEE Trans. Autom. Control*, accepted, to appear.

[5] J. Liu and H. D. Sherali, "A distributed Newton's method for joint multi-hop routing and flow control: Theory and algorithm," in *Proc. IEEE INFOCOM*, Orlando, FL, Mar. 25-30, 2012, pp. 2489–2497.

[6] J. Liu, C. H. Xia, N. B. Shroff, and H. D. Sherali, "Distributed cross-layer optimization in wireless networks: A second-order approach," in *Proc. IEEE INFOCOM*, Turin, Italy, Apr. 14-19, 2013.

[7] A. Pinar, J. Meza, V. Donde, and B. Lesieutre, "Optimization strategies for the vulnerability analysis of the electric power grid," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1786–1810, Jan. 2010.

[8] V. Donde, V. Lopez, B. Lesieutre, A. Pinar, C.Yang, and J. Meza, "Identification of severe multiple contingencies in electric power systems," *IEEE Trans. Power Syst.*, vol. 23, no. 2, pp. 406–417, May 2008.

[9] D. Bienstock and A. Verma, "The $n - k$ problem in power grids: New models, formulations, and numerical experiments," *SIAM Journal of Optimization*, vol. 20, no. 5, pp. 2352–2380, Jun. 2010.

[10] D. Mazauric, S. Soltan, and G. Zussman, "Computational analysis of cascading failures in power networks," in *Proc. ACM Sigmetrics*, Pittsburgh, PA, Jun. 17-21, 2013.

[11] A. Bernstein, D. Bienstock, D. Hay, M. Uzunoglu, and G. Zussman, "Sensitivity analysis of the power grid vulnerability to large-scale cascading failures," *ACM Sigmetrics Performance Evaluation Review*, vol. 40, no. 3, pp. 33–37, Dec. 2012.

[12] D. Bienstock, "Optimal control of cascading power grid failures," in *Proc. IEEE CDC*, Orlando, FL, Dec. 12-15, 2004, pp. 2166–2173.

[13] L. P. Hajdu, J. Peschon, W. F. Tinney, and D. S. Piercy, "Optimal load-shedding policy for power systems," *IEEE Trans. Power App. Syst.*, vol. 87, no. 3, pp. 784–795, Mar. 1968.

[14] H. Seyedi and M. Sanaye-Pasand, "New centralised adaptive load-shedding algorithms to mitigate power system blackouts," *IET Generation, Transmission, and Distribution*, vol. 3, no. 1, pp. 99–114, Jan. 2008.

[15] E. E. Aponte and J. K. Nelson, "Time optimal load shedding for distributed power systems," *IEEE Trans. Power Syst.*, vol. 21, no. 1, pp. 269–277, Feb. 2006.

[16] W. Isamu and N. Makoto, "A genetic algorithm for optimizing switching sequence of service restoration in distribution systems," in *Proc. Congr. Evolutionary Computation*, Jun. 2004, pp. 1683–1690.

[17] W. P. Luan, M. R. Irving, and J. S. Daniel, "Genetic algorithm for supply restoration and optimal load shedding in power system distribution networks," in *Proc. Inst. Elect. Eng., Gen., Transm., Distrib.*, Mar. 2002, pp. 145–151.

[18] S. Toune, H. Fudo, T. Genji, Y. Fukuyama, and Y. Nakanishi, "A reactive Tabu search for service restoration in electric power distribution systems," in *Proc. IEEE World Congr. Computational IntelligenceEvolutionary Computation*, May 1998, pp. 763–768.

[19] H. Mori and Y. Ogita, "A parallel Tabu search based approach to optimal network reconguration for service restoration in distribution systems," in *Proc. Int. Conf. Control Applications*, Sep. 2002, pp. 814–819.

[20] K. H. Jung, H. Kim, and Y. Ko, "Network reconfiguration algorithm for automated distribution systems based on artificial intelligence," *IEEE Trans. Power Del.*, vol. 8, pp. 1933–1941, 1993.

[21] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*, 4th ed. New York: John Wiley & Sons Inc., 2010.

[22] J. J. Grainger and W. D. Stevenson, *Power System Analysis.* New York: McGraw-Hill Inc., 1994.

[23] A. Greenwood, *Electrical Transients in Power Systems.* New York: Wiley-Interscience, 1991.

[24] R. D. Zimmerman, C. E. Murillo-Snchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

[25] J. Lavaei and S. H. Low, "Zero duality gap in optimal power flow problem," *IEEE Trans. Power Syst.*, vol. 27, no. 1, pp. 92–107, Feb. 2012.

[26] J. Garay, S. Kutten, and D. Peleg, "A sub-linear time distributed algorithm for minimum-weight spanning trees (extended abstract)," in *Proc. IEEE Symposium on Foundations of Computer Science (FOCS)*, 1993.

[27] A. Forsgren, P. E. Gill, and M. H. Wright, "Interior methods for nonlinear optimization," *SIAM Review*, vol. 44, no. 4, pp. 525–597, Oct. 2002.

[28] F. R. K. Chung, *Spectral Graph Theory.* Providence, RI: American Mathematical Society, 1994.

[29] Z.-Q. Luo and P. Tseng, "On the linear convergence of descent methods for convex essentially smooth minimization," *SIAM Journal of Control and Optimization*, vol. 30, no. 2, pp. 408–425, Mar. 1992.

[30] R. A. Horn and C. R. Johnson, *Matrix Analysis.* New York, NY: Cambridge University Press, 1990.

[31] "Power systems test case archive." [Online]. Available: http://www.ee.washington.edu/research/pstca/

[32] "City energy consumption and expenditures," The City of Cambridge. [Online]. Available: http://www.cambridgema.gov/~/media/Files/publicworksdepartment/GreenSense/TIP1RESULTS-Final.ashx

# A An Illustrative Example of the SMW Scheme in (23) and Algorithm 1

Here, we give an illustrative example to further explain the remarks for the SMW scheme in (23) and Algorithm 1. Let $l = \sigma(N - i + 1)$ and consider a 3-dimensional symmetric matrix $\mathbf{\Theta}_{(i-1)}^{-1}$ and $(\mathbf{A})_l$ as shown below:

$$\mathbf{\Theta}_{(i)}^{-1} = \begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix} \quad \text{and} \quad (\mathbf{A})_l = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}.$$

This means that Tx($l$) and Rx($l$) are nodes 1 and 3, respectively. Then, it is easy to check that

$$1 + \gamma_l (\mathbf{A})_l^T \mathbf{\Theta}_{(i-1)}^{-1} (\mathbf{A})_l = 1 + af - c^2.$$

That is, $(\mathbf{A})_l$ only picks up elements $a$, $f$, and $c$. Also, note that $1 + af - c^2 > 0$ due to the positive definiteness and hence the diagonal dominance of $\mathbf{\Theta}_{(i-1)}^{-1}$.

Likewise, it is easy to check that

$$\gamma_l \mathbf{\Theta}_{(i-1)}^{-1} (\mathbf{A})_l (\mathbf{A})_l^T \mathbf{\Theta}_{(i-1)}^{-1} = \begin{bmatrix} (a-c)^2 & (a-c)(b-e) & (a-c)(c-f) \\ (a-c)(b-e) & (b-e)^2 & (b-e)(c-f) \\ (a-c)(c-f) & (b-e)(c-f) & (c-f)^2 \end{bmatrix},$$

which means that $(\mathbf{A})_l$ only picks up elements in the 1st and 3rd rows as well as 1st and 3rd columns. Further, since $\mathbf{\Theta}_{i-1}^{-1}$ is symmetric, $(\mathbf{A})_l$ in fact just picks up the elements in 1st and 3rd rows only. Therefore, when doing computation for each line $l$, it suffices for nodes Tx($l$) and Rx($l$) to only store the Tx($l$)-th row and Rx($l$)-th row of $\mathbf{\Theta}_{(i-1)}^{-1}$.

# B Proof of Theorem 5

To prove Theorem 5, we first show the boundedness property of $\mathbf{w}_{[t]}$, i.e., $\|\mathbf{w}_{[t]}\| < \infty$ for all $t$. To prove this result, we have from (40) that

$$\begin{aligned}
&\|\mathbf{w}_{[t]}\| = \|(\mathbf{M}\mathbf{H}_{[t]}^{-1}\mathbf{M}^T)^{-1}[-\mathbf{M}\mathbf{H}_{[t]}^{-1}\nabla g_\mu(\mathbf{y}_{[t]})]\| \\
&\overset{(a)}{\leq} \lambda_{\min}^{-1}\left\{\mathbf{M}\mathbf{H}_{[t]}^{-1}\mathbf{M}^T\right\}\left\|-\mathbf{M}\mathbf{H}_{[t]}^{-1}\nabla g_\mu(\mathbf{y}_{[t]})\right\| \\
&\overset{(b)}{\leq} \lambda_{\min}^{-1}\left\{\mathbf{M}\mathbf{H}_{[t]}^{-1}\mathbf{M}^T\right\}\lambda_{\min}^{-1}\{\mathbf{H}_{[t]}\}\|\mathbf{M}(-\nabla g_\mu(\mathbf{y}_{[t]}))\|, \qquad (44)
\end{aligned}$$

where $(a)$ follows from taking the smallest eigenvalue of $\mathbf{M}\mathbf{H}_{[t]}^{-1}\mathbf{M}^T$ and factoring it outside the norm; and $(b)$ follows from factoring $\lambda_{\min}^{-1}\{\mathbf{H}_{[t]}\}$ outside the norm. By assumption, since $g_\mu(\mathbf{y}_{[t]})$ is Lipschitz continuous, implying that the spectral radius $\rho(\mathbf{H}_{[t]})$ is bounded. Also, since $\mathbf{M}$ is constructed by the node-arc incidence matrix of a connected graph, implying that $\rho(\mathbf{H}_{[t]})$ is finite. As a result, $\lambda_{\min}^{-1}\left\{\mathbf{M}\mathbf{H}_{[t]}^{-1}\mathbf{M}^T\right\}$ must be finite. Therefore, we can conclude that the RHS of (44) is bounded.

With the boundedness of $\mathbf{w}_{[t]}$, we now prove Theorem 5. The main idea and the key steps of the proof are as follows. First, we analyze the one-slot drift of the following quadratic Lyapunov function $V(\mathbf{y}_{[t]}) \triangleq \frac{1}{2}\|\mathbf{y}_{[t]} - \mathbf{y}^*\|^2$, which can be interpreted as measuring the distance between the current iterate $\mathbf{y}_{[t]}$ to the optimal solution $\mathbf{y}^*$. The one-slot drift analysis will reveal the following relationship:

$$\begin{aligned}
\Delta V(y_{[t]}) &\triangleq V(\mathbf{y}_{[t+1]}) - V(\mathbf{y}_{[t]}) \\
&\leq -s^{\min}\|\mathbf{y}_{[t]} - \mathbf{y}^*\| + B_1 + B_2,
\end{aligned}$$

where $s_\mu^{\min} \triangleq \inf_t\{s_{[t]}\}$ represents the lower bound of the step-sizes under $\mu$ and $B_1$ and $B_2$ are some positive constants. Based on this relationship, the result in Theorem 5 follows from telescoping $T$ one-slot drifts and then letting $T$ go to infinity. We now begin with evaluating the one-slot Lyapunov drift $\Delta V(\mathbf{y}_{[t]})$:

$$\begin{aligned}
&\Delta V(\mathbf{y}_{[t]}) = \frac{1}{2}\|\mathbf{y}_{[t+1]} - \mathbf{y}^*\|^2 - \frac{1}{2}\|\mathbf{y}_{[t]} - \mathbf{y}^*\|^2 \\
&= \frac{1}{2}\left(\mathbf{y}_{[t+1]} + y_{[t]} - 2\mathbf{y}^*\right)^T\left(\mathbf{y}_{[t+1]} - \mathbf{y}_{[t]}\right) \\
&= -s_{[t]}(\mathbf{y}_{[t]} - \mathbf{y}^*)\mathbf{H}_{[t]}^{-1}(\nabla g_\mu(\mathbf{y}_{[t]}) + \mathbf{M}^T\mathbf{w}_{[t]}) \qquad (45) \\
&+ \frac{s_{[t]}^2}{2}(\nabla g_\mu(\mathbf{y}_{[t]}) + \mathbf{M}^T\mathbf{w}_{[t]})^T\mathbf{H}_{[t]}^{-2}(\nabla g_\mu(\mathbf{y}_{[t]}) + \mathbf{M}^T\mathbf{w}_{[t]}). \qquad (46)
\end{aligned}$$

In what follows, we will bound the two expressions in (45) and (46), respectively. For (45), we have

$$
\begin{aligned}
(45) &= -s_{[t]}(\mathbf{y}_{[t]} - \mathbf{y}^*)\mathbf{H}_{[t]}^{-1}(\nabla g_\mu(\mathbf{y}_{[t]}) + \mathbf{M}^T\mathbf{w}_{[t]}) \\
&\overset{(a)}{=} -s_{[t]}(\mathbf{y}_{[t]} - \mathbf{y}^*)^T\mathbf{H}_{[t]}^{-1}(\nabla g_\mu(\mathbf{y}_{[t]}) - \nabla g_\mu(\mathbf{y}^*)) \qquad (47) \\
&\quad - s_{[t]}(\mathbf{y}_{[t]} - \mathbf{y}^*)^T\mathbf{H}_{[t]}^{-1}(\mathbf{M}^T\mathbf{w}_{[t]} - \mathbf{M}^T\mathbf{w}^*), \qquad (48)
\end{aligned}
$$

where $(a)$ follows from the fact that $\nabla g_\mu(\mathbf{y}^*) + \mathbf{M}^T\mathbf{w}^* = \mathbf{0}$.

Next, we evaluate (47) and (48) separately. First, by mean-value theorem, we have the following pair of relationships:

$$
\begin{aligned}
g_\mu(\mathbf{y}_{[t]}) &= g_\mu(\mathbf{y}^*) + (\nabla \mathbf{g}_\mu(\mathbf{y}^*))(\mathbf{y}_{[t]} - \mathbf{y}^*) \\
&\quad + \frac{1}{2}(\mathbf{y}_{[t]} - \mathbf{y}^*)^T\mathbf{H}[\widetilde{\mathbf{y}}_1](\mathbf{y}_{[t]} - \mathbf{y}^*), \qquad (49) \\
g_\mu(\mathbf{y}^*) &= g_\mu(\mathbf{y}_{[t]}) + (\nabla \mathbf{g}_\mu(\mathbf{y}_{[t]}))(\mathbf{y}^* - \mathbf{y}_{[t]}) \\
&\quad + \frac{1}{2}(\mathbf{y}^* - \mathbf{y}_{[t]})^T\mathbf{H}[\widetilde{\mathbf{y}}_2](\mathbf{y}^* - \mathbf{y}_{[t]}). \qquad (50)
\end{aligned}
$$

In (49) and (50), $\mathbf{H}[\widetilde{\mathbf{y}}_1]$ and $\mathbf{H}[\widetilde{\mathbf{y}}_2]$ represent the matrices evaluated at points $\widetilde{\mathbf{y}}_1$ and $\widetilde{\mathbf{y}}_2$, where $\widetilde{\mathbf{y}}_1 = (1-\alpha_1)\mathbf{y}_{[t]} + \alpha_1\bar{\mathbf{y}}^*$ and $\widetilde{\mathbf{y}}_2 = (1-\alpha_2)\mathbf{y}_{[t]} + \alpha_2\bar{\mathbf{y}}^*$, for some $0 \le \alpha_1, \alpha_2 \le 1$. Next, adding (49) and (50) yields:

$$
\left(\nabla g_\mu(\mathbf{y}_{[t]}) - \nabla g_\mu(\mathbf{y}^*)\right)^T \left(\bar{\mathbf{y}}^* - \mathbf{y}_{[t]}\right) + \frac{1}{2}\left(\mathbf{y}_{[t]} - \bar{\mathbf{y}}^*\right)^T \left(\mathbf{H}[\widetilde{\mathbf{y}}_1] + \mathbf{H}[\widetilde{\mathbf{y}}_2]\right)\left(\mathbf{y}_{[t]} - \bar{\mathbf{y}}^*\right) = 0,
$$

which further implies that

$$
\begin{aligned}
&\left(\nabla g_\mu(\mathbf{y}_{[t]}) - \nabla g_\mu(\mathbf{y}^*)\right)^T \left(\mathbf{y}_{[t]} - \bar{\mathbf{y}}^*\right) \\
&= \frac{1}{2}\left(\mathbf{y}_{[t]} - \bar{\mathbf{y}}^*\right)^T \left(\mathbf{H}[\widetilde{\mathbf{y}}_1] + \mathbf{H}[\widetilde{\mathbf{y}}_2]\right)\left(\mathbf{y}_{[t]} - \bar{\mathbf{y}}^*\right) \\
&\ge \lambda_{\min}(\mathbf{H}) \left\|\mathbf{y}_{[t]} - \bar{\mathbf{y}}^*\right\|^2 > 0. \qquad (51)
\end{aligned}
$$

where the last inequality follows from the strict convexity of $g_\mu(\cdot)$. Also due to the strict convexity of $g_\mu(\cdot)$, $\mathbf{H}_k$ is positive definite, we have $(\mathbf{y}_{[t]} - \mathbf{y}^*)^T\mathbf{H}_{[t]}^{-1}(\nabla g_\mu(\mathbf{y}_{[t]}) - \nabla g_\mu(\mathbf{y}^*)) > 0$. Note that $s_{[t]} \in (0, 1]$. Let $s^{\min} = \inf_t\{s_{[t]}\}$ (see the main text for the existence proof of $s^{\min}$). It then follows that

$$
\begin{aligned}
(47) &\le -s^{\min}(\mathbf{y}_{[t]} - \mathbf{y}^*)^T\mathbf{H}_{[t]}^{-1}(\nabla g_\mu(\mathbf{y}_{[t]}) - \nabla g_\mu(\mathbf{y}^*)) \\
&\le -\frac{s^{\min}}{\lambda_{\min}(\mathbf{H})}(\mathbf{y}_{[t]} - \mathbf{y}^*)^T(\nabla g_\mu(\mathbf{y}_{[t]}) - \nabla g_\mu(\mathbf{y}^*)). \qquad (52)
\end{aligned}
$$

Combining (52) and (51), we can conclude that

$$
(47) \le -s^{\min}\left\|\mathbf{y}_{[t]} - \bar{\mathbf{y}}^*\right\|^2. \qquad (53)
$$

Now, we evaluate (48):

$$\begin{aligned}(48) &= -s^{\min}(\mathbf{y}_{[t]} - \mathbf{y}^*)^T \mathbf{H}_{[t]}^{-1} \mathbf{M}^T (\mathbf{w}_{[t]} - \mathbf{w}^*) \\ &\leq \frac{s^{\min}}{\lambda_{\min}(\mathbf{H})} \|(\mathbf{y}_{[t]} - \mathbf{y}^*)^T \mathbf{M}^T (\mathbf{w}_{[t]} - \mathbf{w}^*)\| \\ &\leq \frac{s^{\min}}{\lambda_{\min}(\mathbf{H})} \|\mathbf{y}_{[t]} - \mathbf{y}^*\| \|\mathbf{M}^T (\mathbf{w}_{[t]} - \mathbf{w}^*)\| \end{aligned} \tag{54}$$

Since $\mathbf{y}_{[t]}$ is bounded, $\|\mathbf{y}_{[t]} - \mathbf{y}^*\|$ is bounded. Note also that $\mathbf{w}_{[t]}$ is bounded, which implies that $\|\mathbf{M}^T(\mathbf{w}_{[t]} - \mathbf{w}^*)\|$ is bounded. Thus, we define

$$B_1 \triangleq \frac{s^{\min}}{\lambda_{\min}(\mathbf{H})} \sup_t \{\|\mathbf{y}_{[t]} - \mathbf{y}^*\| \|\mathbf{M}^T (\mathbf{w}_{[t]} - \mathbf{w}^*)\|\}. \tag{55}$$

Lastly, we evaluate (46) as follows:

$$\begin{aligned}(46) &= \frac{s_{[t]}^2}{2} (\nabla g_\mu(\mathbf{y}_{[t]}) + \mathbf{M}^T \mathbf{w}_{[t]})^T \mathbf{H}_{[t]}^{-2} (\nabla g_\mu(\mathbf{y}_{[t]}) + \mathbf{M}^T \mathbf{w}_{[t]}) \\ &\overset{(a)}{\leq} \frac{1}{2\lambda_{\min}^2(\mathbf{H})} \|g_\mu(\mathbf{y}_{[t]}) + \mathbf{M}^T \mathbf{w}_{[t]}\|^2 \\ &\overset{(b)}{\leq} \frac{1}{2\lambda_{\min}^2(\mathbf{H})} \|g_\mu(\mathbf{y}_{[t]}) - g_\mu(\mathbf{y}^*) - \mathbf{M}^T \mathbf{w}^* + \mathbf{M}^T \mathbf{w}_{[t]}\|^2 \\ &\overset{(c)}{\leq} \frac{1}{2\lambda_{\min}^2(\mathbf{H})} \left(\|g_\mu(\mathbf{y}_{[t]}) - g_\mu(\mathbf{y}^*)\| + \|\mathbf{M}^T (\mathbf{w}_{[t]} - \mathbf{w}^*)\|\right)^2, \end{aligned}$$

where $(a)$ follows from $s_{[t]} \leq 1$, $(b)$ follows from $\nabla g_\mu(\mathbf{y}^*) + \mathbf{M}^T \mathbf{w}^* = \mathbf{0}$, and $(c)$ follows from triangular inequality. Since $g_\mu(\cdot)$ is Lipschitz continuous, we have $\|g_\mu(\mathbf{y}_{[t]}) - g_\mu(\mathbf{y}^*)\|$ is bounded. Also, since $\|\mathbf{w}_{[t]}\|$ is bounded, we have $\|\mathbf{M}^T(\mathbf{w}_{[t]} - \mathbf{w}^*)\|$ is bounded. As a result, we define

$$\begin{aligned} B_2 \triangleq \frac{1}{2\lambda_{\min}^2(\mathbf{H})} \sup_t & \big(\|g_\mu(\mathbf{y}_{[t]}) - g_\mu(\mathbf{y}^*)\| \\ & + \|\mathbf{M}^T (\mathbf{w}_{[t]} - \mathbf{w}^*)\|\big)^2. \end{aligned} \tag{56}$$

Finally, combining (53), (55), and (56), we have

$$\begin{aligned} \Delta V(y_{[t]}) &\triangleq V(\mathbf{y}_{[t+1]}) - V(\mathbf{y}_{[t]}) \\ &\leq -s^{\min} \|\mathbf{y}_{[t]} - \mathbf{y}^*\| + B_1 + B_2, \end{aligned} \tag{57}$$

Telescoping $T$ one-slot drift expressions for $t = 0, \ldots, T-1$ yields:

$$V(\mathbf{y}_{[T]}) - V(\mathbf{y}_{[0]}) \leq -s^{\min} \sum_{t=0}^{T-1} \|\mathbf{y}_{[t]} - \mathbf{y}^*\|^2 + T(B_1 + B_2). \tag{58}$$

Next, dividing both sides by $Ts^{\min}$, rearranging terms, and taking $T$ to infinity, we have

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \|\mathbf{y}_{[t]} - \mathbf{y}^*\|^2 \le B_\mu^2,$$

where we define $B_\mu^2 \triangleq \frac{B_1+B_2}{s^{\min}}$. Therefore, as $T$ gets large, we have

$$\left|\frac{1}{T}\sum_{t=0}^{T-1}(\mathbf{y}_{[t]} - \mathbf{y}^*)\right| \overset{(a)}{\le} \left(\frac{1}{T}\sum_{t=0}^{T-1} \|\mathbf{y}_{[t]} - \mathbf{y}^*\|^2\right)^{\frac{1}{2}} \le B_\mu,$$

where $(a)$ follows from the triangular inequality and the basic relationship between $l_1$- and $l_2$-norms. Then, the result stated in Theorem 5 follows by taking limsup and liminf, respectively. This completes the proof of Theorem 5.