An Efficient GPU-based Halpern Accelerating Algorithm for Large-scale DC Optimal Power Flow

Qi Wang, Guojun Zhang, Yue Yang, Chao Ren, Wenchuan Wu, *Fellow, IEEE*, Xinyuan Zhao, Mikael Skoglund, *Fellow, IEEE*, and Defeng Sun

***Abstract*—With numerous renewable generators and energy storage systems integrated into the power grids, the security-constrained DC optimal power flow (DCOPF) is essential for power system operation. For large-scale power grids, traditional CPU-based optimization algorithms (such as the simplex and barrier methods) have saturated in computational efficiency and are inherently difficult to parallelize. To tackle these issues, by incorporating the symmetric Gauss–Seidel (sGS) decomposition, this work develops a GPU-based Halpern Peaceman-Rachford algorithm, termed the sGS-HPR, which enjoys an $O(1/k)$ iteration complexity in terms of the KKT residual. Moreover, the closed-form solutions for all subproblems are derived, which only consist of matrix-vector multiplications and vector operations, and thus can be easily parallelized on GPUs. As a consequence, the developed sGS-HPR algorithm enjoys a $O(N_L \times n/\epsilon)$ non-ergodic computational complexity in terms of floating-point operations for obtaining an ϵ-optimal solution measured by the KKT residual for large-scale DCOPF problems, where n represents the variable dimension, and N_L denotes the number of branches in the power grid. Extensive numerical tests on large-scale power grids, reaching up to the 9241-bus PEGASE system, demonstrate the scalability and superior efficiency of the developed GPU-based sGS-HPR algorithm compared to state-of-the-art methods. Notably, the proposed method achieves a 6× speedup compared with Gurobi for large-scale instances. Additionally, for ultra-large-scale cases, Gurobi throws an "out-of-memory" error, while the proposed sGS-HPR algorithm maintains its computational scalability and efficiency.**

***Index Terms*—GPU acceleration, DC optimal power flow, Halpern iteration, Peaceman-Rachford splitting, symmetric Gauss–Seidel decomposition, convergence rate, computational complexity**

I. Introduction

A. Background

DC Optimal Power Flow (DCOPF) is a crucial operational tool for power system operators and serves as the foundational model in many significant optimization problems, such as the economic dispatch [1], [2], market clearing [3], [4], power network planning [5], [6] and security-constrained unit commitment [7], [8]. As a dynamic optimization model, the DCOPF is widely employed for day-ahead and intraday look-ahead power dispatch. The growing integration of renewable generators (RGs) and energy storage systems (ESSs) into power grids introduces several key challenges [9]:

i) As numerous RGs and ESSs integrated into large-scale power grids, the dimension of the variables in the DCOPF grows significantly. Meanwhile, power grid operations exhibit strong spatio-temporal coupling characteristics, substantially increasing the computational burden [10].

ii) With the growing integration of RGs, transmission congestion becomes a critical issue. As a result, there will be more active operational constraints in the DCOPF problems, raising computational effort substantially [11].

Under these circumstances, the intraday look-ahead dispatch imposes exceedingly high requirements on the computational performance. Thus, the DCOPF needs algorithms that exhibit superior computational efficiency, rapid convergence speed and good scalability. Traditional CPU-based methods, such as the interior-point method, are capable of obtaining high-precision solutions. However, their computational complexity grows significantly with problem size. Additionally, their inherently sequential nature limits the benefits that can be gained from modern GPU acceleration. Consequently, solving the DCOPF problem remains certain challenges for large-scale power grids due to the high computational efforts and memory costs. This work tackles these challenges by proposing an efficient and easily parallelizable algorithm, along with its GPU-based implementation.

B. Previous Research

A considerable number of studies have investigated algorithms for solving the DCOPF problem. For instance, six decomposition algorithms were reviewed in [12] for solving the OPF problems, including the analytical target cascading, optimality condition decomposition, alternating direction method of multipliers (ADMM), auxiliary problem principle, consensus+innovations, and proximal message passing. The auxiliary problem principle was employed in [13] to address multi-period distributed DCOPF problems for multiple interconnected power grids. A DCOPF-based market-clearing scheme was proposed in [14] for multi-area power systems, which was solved using the optimality condition decomposition approach. A spatio-temporal decomposition algorithm was introduced in [15] for solving the coordinated economic dispatch (C-ED) of integrated transmission and distribution grids. Through integrating ADMM with a multi-parametric programming projection-based spatial decomposition algorithm, the C-ED

This work of Defeng Sun was supported by the Research Center for Intelligent Operations Research, RGC Senior Research Fellow Scheme No. SRFS2223-5S02, and GRF Project No. 15307822. This work of Wenchuan Wu was supported by the National Science Foundation of China (Grant. U24B6009) and the Beijing Natural Science Foundation (Grant. L243003). This work of Xinyuan Zhao was supported by the National Natural Science Foundation of China (Grant. 12271015). This work of Qi Wang was supported by the Postdoc Matching Fund Scheme under Grant 1-W38P. (*Corresponding author: Defeng Sun, Wenchuan Wu*)

Qi Wang, Guojun Zhang and Defeng Sun are with the Department of Applied Mathematics at The Hong Kong Polytechnic University, China (e-mail: q-yolo.wang@polyu.edu.hk; guojun.zhang@connect.polyu.hk; defeng.sun@polyu.edu.hk). Yue Yang is with the Anhui Provincial Key Laboratory of New Energy Utilization and Energy Saving, Hefei University of Technology, China. (e-mail: yangyuethu@foxmail.com). Chao Ren and Mikael Skoglund are with the KTH Royal Institute of Technology, Sweden (e-mail: renc0003@e.ntu.edu.sg; skoglund@kth.se). Wenchuan Wu is with the Department of Electrical Engineering at Tsinghua University, China (e-mail: wuwench@tsinghua.edu.cn). Xinyuan Zhao is with the School of Mathematics, Statistics and Mechanics at the Beijing University of Technology, China (e-mail: xyzhao@bjut.edu.cn).

problem was addressed in a distributed and parallel manner. A hierarchical distributed algorithm was developed in [16] for the predictive control of a distribution network.

Despite their effectiveness, these approaches face two major computational bottlenecks. First, they are all implemented on CPU platforms, without fully exploiting the parallel processing capabilities that modern hardware, such as GPUs, can offer. Second, they depend on commercial solvers such as Gurobi [17] and CPLEX [18]. While these solvers are highly optimized and efficient for small- to medium-scale problems, applying them to ultra-large-scale problems remains a significant challenge. The primary computational hurdle arises from solving linear systems, which require LU factorization in the simplex method and Cholesky factorization in interior-point methods (IPMs). These factorization-based techniques present two critical drawbacks: (i) excessive memory consumption due to matrix densification during factorization, often resulting in "out-of-memory" errors (OOM) even when the problem instance itself fits into available memory, and (ii) their inherently sequential nature, which hinders parallelization and limits the scalability of GPU acceleration [19]. For example, numerical experiments on 383 MIPLIB instances show that sparse matrix-vector multiplication (SpMV) achieves linear GPU speedup with problem size, whereas Cholesky factorization yields smaller, structure-dependent gains, and $LDL^\top$ factorization provides little to no consistent acceleration (see Figure 1 in [20]).

Recently, some studies have leveraged GPUs to accelerate the solving of DCOPF problems. The authors in [21] proposed a GPU-based proximal message passing algorithm utilizing ADMM for solving contingency-constrained DCOPF problems. A GPU-accelerated gradient-based optimization method was developed in [22]. This method relaxed the equality constraints and incorporated a l_1-norm penalty into the objective function, aiming to satisfy the equality constraints as much as possible. Nevertheless, such an approach can compromise the solution's precision and might result in constraint infeasibility. A primal-dual method was developed in [23] for solving a class of convex optimization problems with affine equality constraints. A distributed optimization method was proposed in [24] for lossy networks using a relaxed ADMM approach, which aligns with the Peaceman-Rachford splitting method applied to the dual problem. For general quadratic/-linear programming problems, the authors in [25] developed a versatile solver utilizing the ADMM framework, named OSQP. Furthermore, reference [26] presented a GPU-based implementation of OSQP [25]. It is noteworthy that most of the studies mentioned above are implemented using ADMM. Despite ADMM's versatility and ease of parallelization, it only exhibits a $O(1/\sqrt{k})$ non-ergodic convergence rate in terms of feasibility violations and objective errors when no acceleration techniques are applied [27], [28].

In the realm of large-scale linear programming, the PDLP solver was introduced in [29], [30], built upon the primal-dual hybrid gradient (PDHG) method [31]. In particular, the cuPDLP.jl [19], a GPU-accelerated version of the PDLP, has demonstrated notable efficiency advantages over traditional commercial solvers such as Gurobi, especially when dealing with large-scale linear programming problems. It is also noteworthy that the iteration complexity of the ergodic sequence from the semi-proximal ADMM method [32], including PDHG [33], [34], is $O(1/k)$ in terms of the feasibility violation and objective errors [28].

More recently, research has focused on accelerating the preconditioned Peaceman-Rachford (PR) method [35], [36], [37] using the Halpern iteration [38], [39]. Specifically, the authors in [35] proposed the Halpern Peaceman-Rachford (HPR) method without proximal terms by integrating the Halpern iteration with PR method [40], [41], achieving an iteration complexity of $O(1/k)$ for the Karush-Kuhn-Tucker (KKT) residual. Later, the authors in [36] reformulated the proximal PR method as a proximal point method (PPM) with a positive definite preconditioner, and developed an HPR method with proximal terms that maintains an iteration complexity of $O(1/k)$ for the weighted fixed-point residual. Further advancements by [37] transformed the semi-proximal PR method into a degenerate PPM (dPPM) with a positive semi-definite preconditioner [42]. By applying the Halpern iteration to this dPPM, they introduced an HPR method with semi-proximal terms, preserving the $O(1/k)$ iteration complexity for the KKT residual. Expanding upon this, a GPU-based semi-proximal HPR method was implemented in [43].

Despite the progress in GPU-accelerated optimization methods, several critical challenges remain from a practical application perspective, which serve as the primary motivation for this work:

i) Given that the DCOPF is of extremely large scale, and shows strong spatio-temporal coupling characteristics, can the existing algorithms, including those implemented on the GPU, meet the computational efficiency requirements?

ii) Most existing GPU-based algorithms are designed for general problems without fully exploiting the problem structure. Can leveraging the problem structure further reduce computational complexity?

C. Contributions

For large-scale power grids, the multi-period DCOPF problem becomes exceedingly complex. The non-zero elements (nnz) in its coefficient matrix can scale up to $\mathbf{10^9}$, posing significant challenges to both computation and memory usage. Under these circumstances, motivated by the recent progress in accelerated algorithms based on Halpern iteration, we propose an efficient GPU-based Halpern Peaceman-Rachford algorithm incorporating the symmetric Gauss–Seidel (sGS) decomposition technique (hereafter named **sGS-HPR**), for large-scale security-constrained DCOPF problems.

The main contributions of this work include:

i) **Algorithm**: Through incorporating the symmetric Gauss–Seidel decomposition technique, we develop the sGS-HPR algorithm, which achieves an iteration complexity of $O(1/k)$ with respect to the KKT residual.

ii) **Implementation**: We then provide the fast implementation of the proposed sGS-HPR algorithm. Specifically, the closed-form solutions are derived for all subproblems, which only consist of matrix-vector multiplications and vector operations, and thus can be easily parallelized on GPUs. As a consequence, the per-iteration

computational complexity of the developed sGS-HPR algorithm in terms of floating-point operations (flops) is $O(T \times N_L \times (N_G + N_{RG} + N_{ESS}))$ for large-scale DCOPF problems, where T, N_L, N_G, N_{RG}, and N_{ESS} represent the optimization horizon, the number of branches, conventional generators, renewable generators, and energy storage systems, respectively. Then, combining the iteration complexity and the per-iteration computational cost, we establish that the proposed algorithm achieves a non-ergodic overall computational complexity of $O(T \times N_L \times (N_G + N_{RG} + N_{ESS})/\epsilon)$ in terms of flops to attain an ϵ-optimal solution, as measured by the KKT residual.

iii) **Effectiveness**: Extensive numerical tests on large-scale power grids, reaching up to the 9241-bus PEGASE system, demonstrate the scalability and superior efficiency of the developed GPU-based sGS-HPR algorithm compared to state-of-the-art methods, the cuPDLP, HPR-LP and Gurobi v12.0.0, especially for large-scale problems. Notably, the proposed method achieves a **6×** speedup compared with Gurobi which operates in its default multi-threaded mode for large-scale instances. Additionally, for ultra-large-scale cases, Gurobi encounters an "out-of-memory" error, whereas the proposed sGS-HPR algorithm maintains its computational scalability and efficiency.

The remainder is organized as follows. We describe the mathematical model of the security-constrained DCOPF problem in Section II. In Section III, we first present the HPR method with semi-proximal terms. Then, we develop the sGS-HPR algorithm by incorporating the sGS decomposition method, along with detailed convergence and iteration complexity analysis. In Section IV, the fast implementation of the sGS-HPR algorithm is provided. We derive the closed-form update expressions for all subproblems and analyze the computational complexity of the developed algorithm. Section V presents the case studies and comparisons. The conclusions are drawn in Section VI.

Notation. Let $\mathbb{R}^n$ be the n-dimensional Euclidean space, which is equipped with the corresponding induced norm $\|\cdot\|$, inner product $\langle\cdot,\cdot\rangle$, and transpose $(\cdot)^{\mathrm{T}}$. We define the nonnegative orthant of $\mathbb{R}^n$ as $\mathbb{R}^n_+$. For a given matrix $A \in \mathbb{R}^{m\times n}$, its transpose is represented as A^*, and the spectral norm of A is expressed as $\|A\| := \sqrt{\lambda_1(AA^*)}$, where $\lambda_1(AA^*)$ denotes the largest eigenvalue of the matrix AA^*. The Kronecker product is denoted by $\otimes$. Moreover, for a self-adjoint positive semi-definite operator $\mathcal{M} : \mathbb{R}^n \to \mathbb{R}^n$, the semi-norm $\|x\|_{\mathcal{M}} := \sqrt{\langle x, \mathcal{M}x\rangle}$ is defined for vector $x \in \mathbb{R}^n$. If $f : \mathbb{R}^n \to (-\infty, +\infty]$ is a convex function, we denote its effective domain as $\mathrm{dom}(f) := \{x \in \mathbb{R}^n \mid f(x) < +\infty\}$, the conjugate of f by $f^*(z) := \sup_{x\in\mathbb{R}^n}\{\langle x, z\rangle - f(x)\}$ for $z \in \mathbb{R}^n$ and its subdifferential by $\partial f(\cdot)$. Moreover, the proximal mapping of f is given by $\mathrm{Prox}_f(x) := \arg\min_{z\in\mathbb{R}^n}\left\{f(z) + \frac{1}{2}\|z - x\|^2\right\}$ for any $x \in \mathbb{R}^n$. Considering a convex set $C \subseteq \mathbb{R}^n$, the indicator function over C is denoted by $\delta_C(\cdot)$, defined as $\delta_C(x) = 0$ if $x \in C$ and $\delta_C(x) = +\infty$ if $x \notin C$. For a closed convex set $C \subseteq \mathbb{R}^n$ and a point $x \in \mathbb{R}^n$, the Euclidean projection of x onto C is formulated as $\Pi_C(x) := \arg\min\{\|x - z\| \mid z \in C\}$. For any $x \in C$, we define the normal cone of C at x as $\mathcal{N}_C(x)$.

II. Mathematical Model Formulation

A. Operational Constraints

In the DCOPF problems, the constraints listed below need to be satisfied, including the power balance constraints, network constraints, spinning reserve constraints, ramping constraints, RG's output constraints, ESS's operational constraints, whose meanings are detailed in [15].

$$\sum_{i\in\mathbb{G}} p_{\mathrm{G},i}(t) + \sum_{i\in\mathbb{RG}} p_{\mathrm{RG},i}(t) + \sum_{i\in\mathbb{E}} \left(p^{\mathrm{dc}}_{\mathrm{ESS},i}(t) - p^{\mathrm{ch}}_{\mathrm{ESS},i}(t)\right) = \sum_{i\in\mathbb{D}} P_{\mathrm{D},i}(t), \quad \forall t \in \mathbb{T}, \tag{1}$$

$$-\bar{P}_{\mathrm{L},j} \le \sum_{i\in\mathbb{G}} SF_{j-i}p_{\mathrm{G},i}(t) + \sum_{i\in\mathbb{RG}} SF_{j-i}p_{\mathrm{RG},i}(t) + \sum_{i\in\mathbb{E}} SF_{j-i}\left(p^{\mathrm{dc}}_{\mathrm{ESS},i}(t) - p^{\mathrm{ch}}_{\mathrm{ESS},i}(t)\right) - \sum_{i\in\mathbb{D}} SF_{j-i}P_{\mathrm{D},i}(t) \le \bar{P}_{\mathrm{L},j}, \quad \forall j \in \mathbb{L}, \quad \forall t \in \mathbb{T}, \tag{2}$$

$$0 \le ru_i(t) \le RU_i\Delta t, \quad 0 \le rd_i(t) \le RD_i\Delta t, \tag{3}$$

$$ru_i(t) \le \bar{P}_{\mathrm{G},i} - p_{\mathrm{G},i}(t), \quad rd_i(t) \le p_{\mathrm{G},i}(t) - \underline{P}_{\mathrm{G},i}, \tag{4}$$

$$\sum_{i\in G} ru_i(t) \ge SRU(t), \quad \sum_{i\in G} rd_i(t) \ge SRD(t), \quad \forall t \in \mathbb{T}, \tag{5}$$

$$-RD_i\Delta t \le p_{\mathrm{G},i}(t) - p_{\mathrm{G},i}(t-1) \le RU_i\Delta t, \quad \forall t \in \mathbb{T}\setminus\{1\}, \tag{6}$$

$$\underline{P}_{\mathrm{RG},i} \le p_{\mathrm{RG},i}(t) \le \bar{P}_{\mathrm{RG},i}, \quad \forall i \in \mathbb{RG}, \forall t \in \mathbb{T}, \tag{7}$$

$$\underline{E}_{\mathrm{ESS},i} \le E_{\mathrm{ESS},i}(0) + \sum_{\tau=1}^{t}\left(p^{\mathrm{ch}}_{\mathrm{ESS},i}(\tau) \times \eta^{\mathrm{ch}}_{\mathrm{ESS},i} - p^{\mathrm{dc}}_{\mathrm{ESS},i}(\tau)/\eta^{\mathrm{dc}}_{\mathrm{ESS},i}\right)\Delta t \le \bar{E}_{\mathrm{ESS},i}, \quad \forall i \in \mathbb{E}, \forall t \in \mathbb{T}, \tag{8}$$

$$\sum_{\tau=1}^{T}\left(p^{\mathrm{ch}}_{\mathrm{ESS},i}(\tau) \times \eta^{\mathrm{ch}}_{\mathrm{ESS},i} - p^{\mathrm{dc}}_{\mathrm{ESS},i}(\tau)/\eta^{\mathrm{dc}}_{\mathrm{ESS},i}\right)\Delta t = 0, \tag{9}$$

$$0 \le p^{\mathrm{ch}}_{\mathrm{ESS},i}(t) \le \bar{P}^{\mathrm{ch}}_{\mathrm{ESS},i}, \quad 0 \le p^{\mathrm{dc}}_{\mathrm{ESS},i}(t) \le \bar{P}^{\mathrm{dc}}_{\mathrm{ESS},i}, \tag{10}$$

where $p_{\mathrm{G},i}(t)$ and $p_{\mathrm{RG},i}(t)$ are the active power outputs of the traditional generator and the RG at bus i . $p^{\mathrm{dc}}_{\mathrm{ESS},i}(t)$ and $p^{\mathrm{ch}}_{\mathrm{ESS},i}(t)$ denote the discharging and charging active power of the ESS at bus i . $P_{\mathrm{D},i}(t)$ represents the active power demand of the load at bus i . $\mathbb{G}$, $\mathbb{RG}$, $\mathbb{E}$, $\mathbb{D}$, $\mathbb{T}$ and $\mathbb{L}$ are the index sets corresponding to the traditional generators, RGs, ESSs, loads, time horizons and lines, respectively. SF_{j-i} represents the shift factor for bus i on line j . $\bar{P}_{\mathrm{L},j}$ denotes the line flow limit of line j. $ru_i(t)$ and $rd_i(t)$ denote the upward and downward spinning reserves of the generator at bus i. RU_i and RD_i are the available upward and downward ramp rates of the generator at bus i. Δt is the optimization time interval. $\bar{P}_{\mathrm{G},i}$ and $\underline{P}_{\mathrm{G},i}$ represent the maximum and minimum active power outputs of the generator at bus i. $SRU(t)$ and $SRD(t)$ are the system-wide requirements for upward and downward spinning reserves. $\bar{P}_{\mathrm{RG},i}$ and $\underline{P}_{\mathrm{RG},i}$ represent the maximum and minimum active power outputs of the RG at bus i. $E_{\mathrm{ESS},i}(0)$, $\underline{E}_{\mathrm{ESS},i}$ and $\bar{E}_{\mathrm{ESS},i}$ are the initial capacity and the capacity limits (both lower and upper) of the ESS at bus i. $\eta^{\mathrm{ch}}_{\mathrm{ESS},i}$ and $\eta^{\mathrm{dc}}_{\mathrm{ESS},i}$ represent the charging and discharging

efficiencies of the ESS at bus i. $\bar{P}^{\text{ch}}_{\text{ESS},i}$ and $\bar{P}^{\text{dc}}_{\text{ESS},i}$ denote the upper bounds on the charging and discharging active power of the ESS at bus i.

Remark 1. *It should be highlighted that the provided model is quite foundational. The goal is to present a versatile model that can be easily integrated into specific scenarios without diminishing the applicability of the proposed method. For instance, the given model plays a foundational role in economic dispatch, market clearing, power network planning, and security-constrained unit commitment.*

Remark 2. *It is also important to emphasize that the proposed method is also well-suited for uncertainty modeling scenarios. By leveraging historical data, existing methods can replace constraints involving stochastic variables with deterministic ones. For instance, the Gaussian Mixture Model approximates the probability distribution of stochastic variables, enabling the transformation of constraints into deterministic equivalents through the chance constraint method at a specified confidence level [44]. Consequently, the stochastic DCOPF problem can be recast as a deterministic DCOPF problem, making the proposed method applicable to large-scale power systems with renewable energy generation.*

B. The Objective Function

The objective of the DCOPF problem is formulated as:

$$\min \left\{ \sum_{t\in\mathbb{T}} \left(\sum_{i\in\mathbb{G}} C_i\left(p_{\text{G},i}(t)\right) + \sum_{i\in\mathbb{RG}} C_i\left(p_{\text{RG},i}(t)\right) + \sum_{i\in\mathbb{E}} C_i\left(P^{\text{dc},}_{\text{ESS},i}(t), P^{\text{ch},}_{\text{ESS},i}(t)\right) \right) \right\}, \tag{11}$$

where $C_i\left(p_{\text{G},i}(t)\right)$ represents the generation cost of the traditional generator. $C_i\left(p_{\text{RG},i}(t)\right)$ and $C_i\left(P^{\text{dc}}_{\text{ESS},i}(t), P^{\text{ch}}_{\text{ESS},i}(t)\right)$ are the curtailment penalization for the RG, and the penalty term for the charging and discharging losses of the ESS.

We employ a linear formulation for the generation cost of thermal power units, which is widely adopted in economic dispatch problems [45], [46], [47], [48] and large-scale security-constrained unit commitment [8], [49].

$$C_i\left(p_{\text{G},i}(t)\right) = a_{1,i}p_{\text{G},i}(t) + a_{0,i}, \tag{12}$$

where $a_{1,i}$ and $a_{0,i}$ denote the coefficients of the linear and constant terms, respectively.

$$C_i\left(p_{\text{RG},i}(t)\right) = \sigma_{\text{RG}}\left(\bar{P}_{\text{RG},i} - p_{\text{RG},i}(t)\right), \tag{13}$$

where σ_{RG} is a positive parameter used to penalize RGs' output curtailments [50], [51].

$$C_i\left(P^{\text{dc}}_{\text{ESS},i}(t), P^{\text{ch}}_{\text{ESS},i}(t)\right) = \sigma_{\text{ESS}}\left(P^{\text{dc}}_{\text{ESS},i}(t)\left(\frac{1}{\eta^{\text{dc}}_{\text{ESS},i}} - 1\right) + P^{\text{ch}}_{\text{ESS},i}(t)\left(1 - \eta^{\text{ch}}_{\text{ESS},i}\right)\right), \tag{14}$$

where σ_{ESS} denotes a positive penalty coefficient [52], [53].

C. The Compact Form of the Above DCOPF Problem

The aforementioned DCOPF optimization problem can be reformulated as the following linear programming problem:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \ & \langle c, x\rangle \\ \text{s.t. } & A_1 x = b_1 \\ & A_2 x \geq b_2 \\ & x \in C, \end{aligned} \tag{15}$$

where the objective function is represented by (11). The equality constraint $A_1x = b_1$ corresponds to (1) and (9). The inequality constraint $A_2x \geq b_2$ includes (2), (4)-(6) and (8). The box constraint $x \in C$ takes into account the upper and lower bounds of the optimization variables. For clarity in later expressions, we set $A_1 \in \mathbb{R}^{m_1\times n}, A_2 \in \mathbb{R}^{m_2\times n}, b_1 \in \mathbb{R}^{m_1}, b_2 \in \mathbb{R}^{m_2}, c \in \mathbb{R}^n$, and $C := \{x \in \mathbb{R}^n \mid l \leq x \leq u\}$ with given vectors $l \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$. Let $A = [A_1; A_2] \in \mathbb{R}^{m\times n}$ with $m = m_1 + m_2$, and $b = [b_1; b_2] \in \mathbb{R}^m$. The constraint dimension m can be expressed as the product of T and a linear combination of N_G, N_L, and N_{ESS}. Similarly, the number of variables n is given by the product of T and a linear combination of N_G, N_{RG}, and N_{ESS}.

Then, the dual form of the problem (15) is formulated as:

$$\begin{aligned} \min_{y\in\mathbb{R}^m, z\in\mathbb{R}^n} \ & -\langle b, y\rangle + \delta_D(y) + \delta^*_C(-z) \\ \text{s.t. } & A^*y + z = c, \end{aligned} \tag{16}$$

where $\delta_D(\cdot)$ denotes the indicator function over $D := \left\{y = (y_1, y_2) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2}_+\right\}$ and $\delta^*_C(\cdot)$ represents the convex conjugate of the indicator function $\delta_C(\cdot)$ of the set C.

In the subsequent derivations, we need to use the invertibility property of $A_1A_1^*$. To ensure this, we present the following proposition, whose proof is provided in ***Appendix A***.

Proposition 1. *A_1 has full row rank.*

III. The Proposed sGS-HPR Algorithm

We begin this section by presenting the HPR method with semi-proximal terms. Then, we derive the sGS-HPR algorithm by employing the sGS decomposition method, along with a comprehensive analysis of its convergence and iteration complexity.

A. An HPR Method

For any $(y, z, x) \in \mathbb{W} := \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n$, the augmented Lagrangian function corresponding to the dual problem (16) is given by

$$\begin{aligned} L_\sigma(y, z; x) := & -\langle b, y\rangle + \delta_D(y) + \delta^*_C(-z) + \\ & \langle x, A^*y + z - c\rangle + \frac{\sigma}{2}\left\|A^*y + z - c\right\|^2, \end{aligned} \tag{17}$$

where $\sigma > 0$ denotes a given penalty parameter.

To simplify the notation, let $w := (y, z, x) \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n$. Then, an HPR method with semi-proximal terms, as proposed in [37] for solving problems (15) and (16) is outlined in Algorithm 1. Essentially, Algorithm 1 is an implementation of the accelerated preconditioned ADMM (pADMM) framework [37] with $\alpha = 2$, where Step 5 employs the Halpern iteration with a stepsize of $\frac{1}{k+2}$ [38], [39]. This formulation is

specifically referred to as the HPR method with semi-proximal terms. For the optimization problem (16), if the proximal terms are removed, the Halpern-accelerated pADMM [37] reduces to the HPR method described in [35].

Algorithm 1 An HPR method for solving the linear programming problem (16).

Set the penalty parameter $\sigma > 0$. Let $\mathcal{T}_1 : \mathbb{R}^m \to \mathbb{R}^m$ be a self-adjoint positive semi-definite linear operator such that $\mathcal{T}_1 + AA^*$ is positive definite. Denote $\bar{w}^k = \left(\bar{y}^k, \bar{z}^k, \bar{x}^k\right)$. Choose an initial point $w^0 = \left(y^0, z^0, x^0\right)$.
for $k = 0, 1, \ldots,$ **do**
  Step 1. $\bar{z}^{k+1} = \underset{z \in \mathbb{R}^n}{\arg\min} \left\{ L_\sigma \left(y^k, z; x^k \right) \right\}$;
  Step 2. $\bar{x}^{k+1} = x^k + \sigma(A^* y^k + \bar{z}^{k+1} - c)$;
  Step 3. $\bar{y}^{k+1} = \underset{y \in \mathbb{R}^m}{\arg\min} \left\{ L_\sigma \left(y, \bar{z}^{k+1}; \bar{x}^{k+1} \right) + \frac{\sigma}{2} \|y - y^k\|^2_{\mathcal{T}_1} \right\}$;
  Step 4. $\hat{w}^{k+1} = 2\bar{w}^{k+1} - w^k$;
  Step 5. $w^{k+1} = \frac{1}{k+2} w^0 + \frac{k+1}{k+2} \hat{w}^{k+1}$;
end for

B. Advancing from the HPR Algorithm to the sGS-HPR Algorithm

Note that the z- and x-update steps in Algorithm 1 both have explicit closed-form solutions that facilitate efficient computation. The computational bottleneck of Algorithm 1 lies in the y-update step, that is, Step 3. The crucial issue is determining the operator $\mathcal{T}_1$ in a way that ensures the computational efficiency of the y-update step while minimizing the total iteration count of Algorithm 1.

Since the optimization problem (15) involves both equality and inequality constraints, in order to simplify the solution of subproblems involving the variable $y = (y_1, y_2)$, we introduce a specific operator called the symmetric Gauss-Seidel operator [54], [55]. Its core idea is to segregate equality constraints and inequality constraints, which also separates the smooth and non-smooth terms in the objective function.

Specifically, we first define a self-adjoint, symmetric, positive semi-definite linear operator $\mathcal{H} : \mathbb{R}^m \to \mathbb{R}^m$ as follows:

$$\mathcal{H} := \begin{bmatrix} A_1 A_1^* & A_1 A_2^* \\ A_2 A_1^* & A_2 A_2^* + \mathcal{S}_2 \end{bmatrix}, \tag{18}$$

where $\mathcal{S}_2 : \mathbb{R}^{m_2} \to \mathbb{R}^{m_2}$ is a self-adjoint positive semi-definite linear operator such that $A_2 A_2^* + \mathcal{S}_2$ is symmetric positive definite.

Then, the operator $\mathcal{H}$ can be decomposed as follows:

$$\mathcal{H} = U_{\mathcal{H}}^* + D_{\mathcal{H}} + U_{\mathcal{H}}, \tag{19}$$

where

$$U_{\mathcal{H}} = \begin{bmatrix} 0 & A_1 A_2^* \\ 0 & 0 \end{bmatrix}, \quad D_{\mathcal{H}} = \begin{bmatrix} A_1 A_1^* & 0 \\ 0 & A_2 A_2^* + \mathcal{S}_2 \end{bmatrix}.$$

Furthermore, the sGS operator corresponding to $\mathcal{H}$ denoted by $\mathrm{sGS}(\mathcal{H}) : \mathbb{R}^m \to \mathbb{R}^m$ is defined as follows [55]:

$$\mathrm{sGS}(\mathcal{H}) = U_{\mathcal{H}}^* D_{\mathcal{H}}^{-1} U_{\mathcal{H}}. \tag{20}$$

As a result, the operator $\mathcal{T}_1$ can be defined in (21), under which setting $\bar{y_1}^{k+1}$ and $\bar{y_2}^{k+1}$ can be efficiently computed separately, as shown in Proposition 2. The proof of Proposition 2 is detailed in ***Appendix B***.

$$\mathcal{T}_1 = \mathrm{sGS}(\mathcal{H}) + \mathrm{diag}(\mathbf{0}_{m_1}, \mathcal{S}_2). \tag{21}$$

Proposition 2. *Combined with (18)-(21), the optimal solution $\bar{y}^{k+1}$ can be computed exactly via the following steps:*

$$\begin{cases} \bar{y}_1^{k+1/2} = \underset{y_1 \in \mathbb{R}^{m_1}}{\arg\min} \{ L_\sigma \left(y_1, y_2^k, \bar{z}^{k+1}; \bar{x}^{k+1} \right) \}, \\ \bar{y}_2^{k+1} = \underset{y_2 \in \mathbb{R}^{m_2}}{\arg\min} \Big\{ L_\sigma (\bar{y}_1^{k+1/2}, y_2, \bar{z}^{k+1}; \bar{x}^{k+1}) + \frac{\sigma}{2} \|y_2 - y_2^k\|^2_{\mathcal{S}_2} \Big\}, \\ \bar{y}_1^{k+1} = \underset{y_1 \in \mathbb{R}^{m_1}}{\arg\min} \left\{ L_\sigma \left(y_1, \bar{y}_2^{k+1}, \bar{z}^{k+1}; \bar{x}^{k+1} \right) \right\}. \end{cases} \tag{22}$$

Moreover, we can obtain that

$$\mathcal{T}_1 + AA^* \succ 0. \tag{23}$$

Finally, the sGS-HPR algorithm for solving the DCOPF problem is presented in Algorithm 2:

Algorithm 2 The sGS-HPR algorithm for solving the DCOPF problems.

Set the penalty parameter $\sigma > 0$. Choose an initial point $w^0 = \left(y^0, z^0, x^0\right) \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n$.
for $k = 0, 1, \ldots,$ **do**
  Step 1. $\bar{z}^{k+1} = \underset{z \in \mathbb{R}^n}{\arg\min} \left\{ L_\sigma \left(y_1^k, y_2^k, z; x^k \right) \right\}$;
  Step 2. $\bar{x}^{k+1} = x^k + \sigma(A_1^* y_1^k + A_2^* y_2^k + \bar{z}^{k+1} - c)$;
  Step 3.1. $\bar{y}_1^{k+1/2} = \underset{y_1 \in \mathbb{R}^{m_1}}{\arg\min} \{ L_\sigma \left(y_1, y_2^k, \bar{z}^{k+1}; \bar{x}^{k+1} \right) \}$;
  Step 3.2. $\bar{y}_2^{k+1} = \underset{y_2 \in \mathbb{R}^{m_2}}{\arg\min} \left\{ L_\sigma \left(\bar{y}_1^{k+1/2}, y_2, \bar{z}^{k+1}; \bar{x}^{k+1} \right) + \frac{\sigma}{2} \|y_2 - y_2^k\|^2_{\mathcal{S}_2} \right\}$;
  Step 3.3. $\bar{y}_1^{k+1} = \underset{y_1 \in \mathbb{R}^{m_1}}{\arg\min} \left\{ L_\sigma \left(y_1, \bar{y}_2^{k+1}, \bar{z}^{k+1}; \bar{x}^{k+1} \right) \right\}$;
  Step 4. $\hat{w}^{k+1} = 2\bar{w}^{k+1} - w^k$;
  Step 5. $w^{k+1} = \frac{1}{k+2} w^0 + \frac{k+1}{k+2} \hat{w}^{k+1}$;
end for

C. Convergence Properties of the sGS-HPR Algorithm

As shown in [56, Corollary 28.3.1], $(y^*, z^*) \in \mathbb{R}^m \times \mathbb{R}^n$ is an optimal solution to problem (16) provided that there exists $x^* \in \mathbb{R}^n$ for which the KKT conditions below hold for (y^*, z^*, x^*):

$$\begin{aligned} 0 &\in Ax^* - b + \mathcal{N}_D(y^*), \\ 0 &\in z^* + \mathcal{N}_C(x^*), \\ A^* y^* + z^* - c &= 0. \end{aligned} \tag{24}$$

Subsequently, for the purpose of analyzing the global convergence of the sGS-HPR algorithm in Algorithm 2, we introduce the following assumption:

Assumption 1. *There exists a vector $(y^*, z^*, x^*) \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n$ that fulfills the KKT system (24).*

With Assumption 1, solving the primal problem (15) and the dual problem (16) amounts to finding a $w^* \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n$

such that $0 \in \mathcal{T}w^*$, with the maximal monotone operator $\mathcal{T}$ given by

$$\mathcal{T}w = \begin{pmatrix} -b + \mathcal{N}_D(y) + Ax \\ -\partial\delta_C^*(-z) + x \\ c - A^*y - z \end{pmatrix}, \quad \forall w = (y, z, x) \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n. \tag{25}$$

Besides, we define the self-adjoint linear operator $\mathcal{M} : \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n$ as follows:

$$\mathcal{M} = \begin{bmatrix} \sigma(AA^* + \mathcal{T}_1) & 0 & A \\ 0 & 0 & 0 \\ A^* & 0 & \frac{1}{\sigma} I_n \end{bmatrix}, \tag{26}$$

where I_n denotes the identity matrix in $\mathbb{R}^{n\times n}$.

Subsequently, we can establish the equivalence between the sGS-HPR algorithm and the accelerated degenerate proximal point method (dPPM) [37], [42] as shown in the following proposition, whose proof is provided in ***Appendix C***.

Proposition 3. *With the operators $\mathcal{T}$ in (25) and $\mathcal{M}$ in (26), the sequence $\{w^k\}$ generated by the sGS-HPR algorithm is equivalent to the sequence $\{w^k\}$ generated by the following accelerated dPPM for any $k \geq 0$,*

$$\begin{cases} \bar{w}^{k+1} = (\mathcal{M} + \mathcal{T})^{-1}\mathcal{M}w^k \\ \hat{w}^{k+1} = 2\bar{w}^{k+1} - w^k \\ w^{k+1} = \dfrac{1}{k+2}w^0 + \dfrac{k+1}{k+2}\hat{w}^{k+1} \end{cases} \tag{27}$$

with the same initial point $w^0 \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n$. Additionally, $\mathcal{M}$ serves as an admissible preconditioner[1] such that $(\mathcal{M} + \mathcal{T})^{-1}$ is Lipschitz continuous.

With the equivalence presented in Proposition 3, the global convergence of the sGS-HPR method in Algorithm 2 can be established by exploring the global convergence properties of dPPM, as outlined in the following proposition, whose proof is provided in ***Appendix D***.

Proposition 4. *Given that Assumption 1 holds, then the sequence $\{\bar{w}^k\} = \{(\bar{y}^k, \bar{z}^k, \bar{x}^k)\}$ generated by the sGS-HPR algorithm converges to the point $w^* = (y^*, z^*, x^*)$, where (y^*, z^*) is the solution of the problem (16) and x^* is the solution to the problem (15).*

Next, the iteration complexity of the sGS-HPR algorithm is analyzed. We define the residual mapping associated with the KKT system (24) in the form of Equation (28), as introduced in [57]. Note that $\mathcal{R}(w) = 0$ implies that $x^* \in \mathbb{R}^n$ and $(y^*, z^*) \in \mathbb{R}^m \times \mathbb{R}^n$ are the solution to problems (15) and (16), respectively.

$$\mathcal{R}(w) = \begin{pmatrix} y - \Pi_D(y - Ax + b) \\ x - \Pi_C(x - z) \\ c - A^*y - z \end{pmatrix}, \quad \forall w = (y, z, x) \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n. \tag{28}$$

[1]In [42], for $\mathcal{T}$, an admissible preconditioner is an operator $\mathcal{M}$ that is linear, bounded, self-adjoint, and positive semidefinite, satisfying the criterion that $(\mathcal{M} + \mathcal{T})^{-1}\mathcal{M}$ is single-valued and has full domain.

Building upon the equivalence in Proposition 3, and incorporating the iteration complexity of the Halpern iteration [39], the following iteration complexity for the sGS-HPR algorithm is derived, whose proof is provided in ***Appendix E***.

Theorem 1. *Provided that Assumption 1 holds, let $\{\bar{w}^k\} = \{(\bar{y}^k, \bar{z}^k, \bar{x}^k)\}$ be the sequence generated by the sGS-HPR method in Algorithm 2, and let $w^* = (y^*, z^*, x^*)$ be the limit point of the sequence $\{(\bar{y}^k, \bar{z}^k, \bar{x}^k)\}$. Define $R_0 = \|w^0 - w^*\|_{\mathcal{M}}$. Then, the sequences $\{w^k\}$ and $\{\hat{w}^k\}$ generated by the sGS-HPR algorithm satisfy the following:*

$$\|w^k - \hat{w}^{k+1}\|_{\mathcal{M}} \leq \frac{2R_0}{k+1}, \quad \forall k \geq 0, \quad w^* \in \mathcal{T}^{-1}(0). \tag{29}$$

Moreover, for all $k \geq 0$, the following iteration complexity bound hold:

$$\left\|\mathcal{R}\left(\bar{w}^{k+1}\right)\right\| \leq \left(\frac{\sigma\left(\|A^*\| + \left\|\sqrt{\mathcal{T}_1}\right\|\right) + 1}{\sqrt{\sigma}}\right)\frac{R_0}{(k+1)}. \tag{30}$$

Remark 3. *It is noteworthy that, in the absence of acceleration, the ADMM has a non-ergodic rate of $O(1/\sqrt{k})$ in terms of KKT residuals [28]. In contrast, the developed sGS-HPR algorithm attains a $O(1/k)$ non-ergodic convergence rate in terms of KKT residuals. This provides the proposed method with significant advantages in solving large-scale DCOPF problems.*

IV. Fast Implementation of the sGS-HPR Algorithm

In this section, we obtain the closed-form update expressions for all subproblems and conduct a computational complexity analysis of the developed sGS-HPR algorithm.

A. The Closed-form Update Expressions for All Subproblems

1) The z- and x-update steps

Employing the expression of the augmented Lagrangian function, as provided in (17), and expanding Step 1 of Algorithm 2, we can derive:

$$\begin{aligned} \bar{z}^{k+1} &= \arg\min_z \left\{L_\sigma\left(y^k, z; x^k\right)\right\} \\ &= \arg\min_z \left\{\delta_C^*(-z) + \langle x^k, A^*y^k + z - c\rangle \right. \\ &\quad \left. + \frac{\sigma}{2}\|A^*y^k + z - c\|^2\right\}. \end{aligned} \tag{31}$$

The optimality condition for the above subproblem is given as follows:

$$0 \in x^k + \sigma\left(A^*y^k + z - c\right) + \partial\delta_C^*(-z). \tag{32}$$

Furthermore, it follows that

$$\bar{z}^{k+1} = \frac{1}{\sigma}\left\{\Pi_C\left[x^k + \sigma\left(A^*y^k - c\right)\right] - \left[x^k + \sigma\left(A^*y^k - c\right)\right]\right\}. \tag{33}$$

Additionally, for the x-update step, we have:

$$\bar{x}^{k+1} = x^k + \sigma\left(A^*y^k + \bar{z}^{k+1} - c\right). \tag{34}$$

2) The y_1-update step

Similarly, by combining (17), we can obtain

$$\begin{aligned}\bar{y}_1^{k+\frac{1}{2}} &= \underset{y_1}{\operatorname{argmin}} L_\sigma\left(y_1, y_2^k, \bar{z}^{k+1}, \bar{x}^{k+1}\right) \\ &= \underset{y_1}{\arg\min}\Big\{-\langle b_1, y_1\rangle + \delta_{\mathbb{R}^{m_1}}(y_1) + \left\langle \bar{x}^{k+1}, A_1^* y_1\right\rangle \\ &+ \frac{\sigma}{2}\left\|A_1^* y_1 + A_2^* y_2^k + \bar{z}^{k+1} - c\right\|^2\Big\}.\end{aligned} \tag{35}$$

The optimal condition for the aforementioned subproblem can be stated as:

$$A_1 A_1^* y_1 = R_1, \tag{36}$$

where $R_1 = \frac{1}{\sigma}\left[b_1 - A_1\left(\bar{x}^{k+1} + \sigma\left(A_2^* y_2^k + \bar{z}^{k+1} - c\right)\right)\right]$.

Leveraging the DCOPF structure and combining the expression of A_1 presented in (55), $A_1 A_1^*$ can be generally expressed in the following block form:

$$A_1 A_1^* = \begin{bmatrix} D_1 & 1_{m_{11}} \otimes d^{\mathrm{T}} \\ 1_{m_{11}}^{\mathrm{T}} \otimes d & D_2 \end{bmatrix}, \tag{37}$$

where $A_1 \in \mathbb{R}^{m_1 \times n}$, $D_1 \in \mathbb{R}^{m_{11} \times m_{11}}$, $D_2 \in \mathbb{R}^{m_{12} \times m_{12}}$, $d \in \mathbb{R}^{m_{12} \times 1}$, $m_{11} + m_{12} = m_1$. $1_{m_{11}}$ is a column vector of dimension m_{11}, with all entries equal to 1. D_1 and D_2 are both diagonal matrices.

Here, y_1 can be further split into two parts, y_{11} and y_{12}.

$$\begin{aligned}&A_1 A_1^* y_1 = R_1 \\ &\Rightarrow \begin{bmatrix} D_1 & 1_{m_{11}} \otimes d^{\mathrm{T}} \\ 1_{m_{11}}^{\mathrm{T}} \otimes d & D_2 \end{bmatrix} \begin{bmatrix} y_{11} \\ y_{12} \end{bmatrix} = \begin{bmatrix} R_{11} \\ R_{12} \end{bmatrix}.\end{aligned} \tag{38}$$

Furthermore, the explicit expressions for y_{11} and y_{12} are given in Proposition 5.

Proposition 5. *The solution y_1 to (38) is given by*

$$\begin{aligned}y_{11} &= \widetilde{y_{11}} - \frac{1}{\frac{1}{d^T D_2^{-1} d} + 1_{m_{11}}^T D_1^{-1} 1_{m_{11}}} \cdot D_1^{-1} 1_{m_{11}} 1_{m_{11}}^T \widetilde{y_{11}}, \\ y_{12} &= D_2^{-1}\left(R_{12} - \left(1_{m_{11}}^T \otimes d\right) y_{11}\right),\end{aligned} \tag{39}$$

where D_1 and D_2 are both diagonal matrices, enabling the above formulas to be computed efficiently. Moreover, $\widetilde{y_{11}}$ is expressed as:

$$\widetilde{y_{11}} = D_1^{-1}\left(R_{11} - \left(\left(1_{m_{11}} \otimes d^T\right) D_2^{-1}\right) R_{12}\right). \tag{40}$$

Proof: Based on (38), we have

$$\begin{aligned}&\begin{bmatrix} D_1 & 1_{m_{11}} \otimes d^{\mathrm{T}} \\ 1_{m_{11}}^{\mathrm{T}} \otimes d & D_2 \end{bmatrix} \begin{bmatrix} y_{11} \\ y_{12} \end{bmatrix} = \begin{bmatrix} R_{11} \\ R_{12} \end{bmatrix} \\ &\Rightarrow \left[D_1 - \left(1_{m_{11}} \otimes d^{\mathrm{T}}\right) D_2^{-1}\left(1_{m_{11}}^{\mathrm{T}} \otimes d\right)\right] y_{11} = \widetilde{R_{11}},\end{aligned} \tag{41}$$

where $\widetilde{R_{11}} = R_{11} - \left(\left(1_{m_{11}} \otimes d^{\mathrm{T}}\right) D_2^{-1}\right) R_{12}$.

By leveraging the properties of the Kronecker product, we can obtain:

$$\begin{aligned}&\left(\left(1_{m_{11}} \otimes d^{\mathrm{T}}\right) D_2^{-1}\right)\left(1_{m_{11}}^{\mathrm{T}} \otimes d\right) \\ &= \left(1_{m_{11}} \otimes d^{\mathrm{T}}\right)\left(1 \otimes D_2^{-1}\right)\left(1_{m_{11}}^{\mathrm{T}} \otimes d\right) \\ &= \left(1_{m_{11}} 1_{m_{11}}^{\mathrm{T}} \otimes d^{\mathrm{T}} D_2^{-1} d\right) = \alpha 1_{m_{11}} 1_{m_{11}}^{\mathrm{T}},\end{aligned} \tag{42}$$

where $\alpha = d^{\mathrm{T}} D_2^{-1} d$ is a constant.

Furthermore, (41) can be reformulated as:

$$\left(D_1 - \alpha 1_{m_{11}} 1_{m_{11}}^{\mathrm{T}}\right) y_{11} = \widetilde{R_{11}}. \tag{43}$$

Based on the Woodbury matrix identity, the inverse of $\left(D_1 - \alpha 1_{m_{11}} 1_{m_{11}}^{\mathrm{T}}\right)$ can be expressed as follows:

$$\begin{aligned}&\left(D_1 - \alpha 1_{m_{11}} 1_{m_{11}}^{\mathrm{T}}\right)^{-1} \\ &= D_1^{-1} - D_1^{-1} 1_{m_{11}}\left(\frac{1}{\alpha} + 1_{m_{11}}^{\mathrm{T}} D_1^{-1} 1_{m_{11}}\right)^{-1} 1_{m_{11}}^{\mathrm{T}} D_1^{-1} \\ &= D_1^{-1} - \frac{1}{\frac{1}{d^{\mathrm{T}} D_2^{-1} d} + 1_{m_{11}}^{\mathrm{T}} D_1^{-1} 1_{m_{11}}} \cdot D_1^{-1} 1_{m_{11}} 1_{m_{11}}^{\mathrm{T}} D_1^{-1}.\end{aligned} \tag{44}$$

Thus, we have

$$\begin{aligned}y_{11} &= \left(D_1 - \alpha 1_{m_{11}} 1_{m_{11}}^{\mathrm{T}}\right)^{-1} \widetilde{R_{11}} \\ &= \left(D_1^{-1} - \frac{1}{\frac{1}{d^{\mathrm{T}} D_2^{-1} d} + 1_{m_{11}}^{\mathrm{T}} D_1^{-1} 1_{m_{11}}} \cdot D_1^{-1} 1_{m_{11}} 1_{m_{11}}^{\mathrm{T}} D_1^{-1}\right) \\ &\quad \left(R_{11} - \left(\left(1_{m_{11}} \otimes d^{\mathrm{T}}\right) D_2^{-1}\right) R_{12}\right).\end{aligned} \tag{45}$$

Moreover, we can obtain y_{12} by

$$\begin{aligned}&D_2 y_{12} = R_{12} - \left(1_{m_{11}}^{\mathrm{T}} \otimes d\right) y_{11} \\ &\Rightarrow y_{12} = D_2^{-1}\left(R_{12} - \left(1_{m_{11}}^{\mathrm{T}} \otimes d\right) y_{11}\right).\end{aligned} \tag{46}$$

This completes the proof.

■

3) The y_2-update step

Before providing the specific closed-form solution of the y_2-update step, we set $\mathcal{S}_2$ in the form given by equation (47). In this way, the optimal solution $\bar{y}_2^{k+1}$ can be computed efficiently.

$$\mathcal{S}_2 = \lambda I_{m_2} - A_2 A_2^* \tag{47}$$

with λ being the largest eigenvalue of $A_2 A_2^*$.

From Step 3.2 of Algorithm 2, we can derive:

$$\begin{aligned}\bar{y}_2^{k+1} &= \underset{y_2}{\operatorname{argmin}}\left(L_\sigma\left(\bar{y}_1^{k+\frac{1}{2}}, y_2, \bar{z}^{k+1}, \bar{x}^{k+1}\right) + \frac{\sigma}{2}\left\|y_2 - y_2^k\right\|_{\mathcal{S}_2}^2\right) \\ &= \underset{y_2}{\arg\min}\Big(-\langle b_2, y_2\rangle + \delta_{\mathbb{R}_+^{m_2}}(y_2) + \left\langle \bar{x}^{k+1}, A_2^* y_2\right\rangle \\ &\quad + \frac{\sigma}{2}\left\|A_1^* \bar{y}_1^{k+\frac{1}{2}} + A_2^* y_2 + \bar{z}^{k+1} - c\right\|^2 \\ &\quad + \frac{\sigma}{2}\left\|y_2 - y_2^k\right\|_{\mathcal{S}_2}^2\Big).\end{aligned} \tag{48}$$

The optimality condition for the above subproblem is given as follows:

$$\begin{aligned}0 \in &-b_2 + \partial\delta_{\mathbb{R}_+^{m_2}}(y_2) + A_2 \bar{x}^{k+1} \\ &+ \sigma A_2\left(A_1^* \bar{y}_1^{k+\frac{1}{2}} + A_2^* y_2 + \bar{z}^{k+1} - c\right) + \sigma\mathcal{S}_2\left(y_2 - y_2^k\right).\end{aligned} \tag{49}$$

Building upon this, we can obtain

$$\bar{y}_2^{k+1} = \Pi_{\mathbb{R}_+^{m_2}}\left[y_2^k + \frac{1}{\lambda}\left(\frac{b_2}{\sigma} - A_2 R_y\right)\right], \tag{50}$$

where $R_y = \frac{\bar{x}^{k+1}}{\sigma} + \left(A_1^* \bar{y}_1^{k+\frac{1}{2}} + A_2^* y_2^k + \bar{z}^{k+1} - c\right)$.

Remark 4. *To exploit the parallel computing capabilities of GPUs, we design custom CUDA kernels to solve all subproblems. Matrix–vector multiplications are carried out using the* `cusparseSpMV()` *routine from the* CUSPARSE *library,*

with the `CUSPARSE_SPMV_CSR_ALG2` algorithm selected to guarantee deterministic results. The coefficient matrices are stored in the Compressed Sparse Row (CSR) format. Each CUDA kernel is configured with $\lceil n/256 \rceil$ blocks and 256 threads per block, ensuring all n elements are processed in parallel.

B. Computational Complexity Analysis

Based on the explicit formulas in (39) and the expression of R_1, the complexity result for solving the linear equation $A_1 A_1^* y_1 = R_1$ can be derived.

Corollary 1. *The linear system $A_1 A_1^* y_1 = R_1$ can be solved in $O(T \times (N_G + N_{RG} + N_{ESS}))$ flops.*

Based on Corollary 1, the per-iteration computational cost of the proposed sGS-HPR method can be determined.

Corollary 2. *For large-scale power systems, the per-iteration computational complexity of Algorithm 2 in terms of flops is $O(T \times N_L \times (N_G + N_{RG} + N_{ESS}))$.*

Combining the iteration complexity presented in Theorem 1 and the per-iteration computational cost outlined in Corollary 2, the following overall computational complexity result of the sGS-HPR method in Algorithm 2 for solving the problem (15) can be derived.

Theorem 2. *Let $\{\bar{y}^k, \bar{z}^k, \bar{x}^k\}$ be the sequence generated by the sGS-HPR method in Algorithm 2. For any given tolerance $\varepsilon > 0$, Algorithm 2 requires at most*

$$\left\lceil \frac{1}{\varepsilon} \left(\frac{\sigma\left(\|A^*\| + \left\|\sqrt{\mathcal{T}_1}\right\|\right) + 1}{\sqrt{\sigma}} R_0 \right) - 1 \right\rceil \tag{51}$$

iterations to yield a solution satisfying $\left\|\mathcal{R}\left(\bar{w}^k\right)\right\| \leq \varepsilon$, where (x^*, z^*) is the limit point of the sequence $\{\bar{x}^k, \bar{z}^k\}$. Particularly, the overall computational complexity of the sGS-HPR method in Algorithm 2 to reach this level of accuracy in terms of flops is

$$O\left(\left(\frac{\sigma\left(\|A^*\| + \left\|\sqrt{\mathcal{T}_1}\right\|\right) + 1}{\sqrt{\sigma}} R_0 \right) \frac{T \times N_L \times (N_G + N_{RG} + N_{ESS})}{\varepsilon} \right). \tag{52}$$

Proof: From (30), we know that the maximum number of iterations for Algorithm 2 can be calculated as:

$$\begin{aligned} &\left(\frac{\sigma\left(\|A^*\| + \left\|\sqrt{\mathcal{T}_1}\right\|\right) + 1}{\sqrt{\sigma}} \right) \frac{R_0}{(k+1)} \leq \varepsilon \\ \Rightarrow\ & k \geq \frac{1}{\varepsilon} \left(\frac{\sigma\left(\|A^*\| + \left\|\sqrt{\mathcal{T}_1}\right\|\right) + 1}{\sqrt{\sigma}} R_0 \right) - 1. \end{aligned} \tag{53}$$

Furthermore, combining the iteration complexity presented in Theorem 1 and the per-iteration computational cost outlined in Corollary 2, the overall computational complexity of the sGS-HPR algorithm can be derived as expressed in (52).

This completes the proof. ■

V. Numerical Experiments

In this section, we comprehensively compare the performance of the proposed sGS-HPR algorithm with state-of-the-art methods on the DCOPF problem, including the cuPDLP algorithm [19], HPR-LP method [43] and commercial solver Gurobi v12.0.0 [17].

A. Experimental Setup

- **Software and computing environment**. The GPU-based algorithms are run on an NVIDIA A100-SXM4-80GB GPU with CUDA 12.3. Gurobi runs on a workstation equipped with an Intel i9-14900HX CPU with 24 cores. The total memory of the workstation is 96 GB. To ensure relative fairness during testing, the maximum memory allocated for Gurobi is set to 80 GB.
- **Preconditioning**. For sGS-HPR, we first apply 10 iterations of Ruiz scaling [58], then adopt the bidiagonal preconditioning proposed by Pock and Chambolle [59] with $\alpha = 1$. Finally, we normalize b and c by dividing them by $b + 1$ and $c + 1$, respectively. The similar preconditioning method is applied in HPR-LP [43]. For cuPDLP [19], the default preconditioning setting is adopted. For Gurobi, keep the parameters related to scaling at their default settings.
- **Initialization and parameter setting**. We initialize sGS-HPR with a cold start and set the initial penalty parameter $\sigma = 1$. Accordingly, the other methods used for benchmarking are initialized with the same initial values. It is worth noting that the parameter σ plays a crucial role in the performance of cuPDLP, HPR-LP and the developed sGS-HPR algorithm. In this work, we adopt an adaptive σ update scheme derived from the complexity results of the sGS-HPR algorithm, as established in Theorem 1, similar to that given in [43]. To balance feasibility accuracy, algorithmic simplicity, and the total number of iterations, we further incorporate a restart strategy [29] into the algorithmic implementation. Notably, Theorem 1 also provides a theoretical bound that guides the design of the restart criteria. The restart criteria are checked every 100 iterations. As these aspects do not constitute an original contributions of this work, we do not elaborate on them here, and please refer to [29], [43] for further details. For cuPDLP, the default σ update strategy is adopted.
- **Termination criteria**. For both sGS-HPR and HPR-LP, the sequence $\{\bar{w}^k\}$ is employed to check the stopping criteria. The algorithms are terminated when the following stopping criteria meet for the given tolerance $\varepsilon = 5 \times 10^{-5}$:

$$\begin{aligned} &\|\Pi_D(b - Ax)\| \leq \varepsilon(1 + \|b\|), \\ &\|x - \Pi_C(x - z)\| \leq \varepsilon(1 + \|x\| + \|z\|), \\ &\|c - A^* y - z\| \leq \varepsilon(1 + \|c\|). \end{aligned} \tag{54}$$

For cuPDLP , we utilize the default stopping criteria. For Gurobi, the parameters OptimalityTol, FeasibilityTol and BarConvTol (for the barrier methods) are set to 5×10^{-5} accordingly.

- **Testing data**. Extensive experiments have been carried out on three case systems of different scales to evaluate the performance of multiple algorithms. Specific details of each case are provided in Table I. Furthermore, to investigate the impact of the growth of the time dimension on the algorithms' performances, we conduct tests on different cases across time scales ranging from 4 to 96. Correspondingly, the number of nonzeros (nnz) in the constraint matrix A covers a range of orders of magnitude, from 10^6 to 10^9, as shown in Table II.

TABLE I
DETAILS OF THE TEST CASES

Cases	Num. of Buses	Num. of Branches	Num. of Generators	Num. of RGs	Num. of ESSs
case1354pegase	1,354	1,991	260	136	68
case2868rte	2,868	3,808	600	286	143
case9241pegase	9,241	16,049	1,445	920	460

- **Time limit**. A time limit of 3600 seconds is imposed for all the cases.

B. Comparisons of the Proposed sGS-HPR Algorithm with Gurobi, cuPDLP and HPR-LP

First, we compare the computational time of the proposed sGS-HPR algorithm with that of Gurobi. Notably, Gurobi incorporates three efficient algorithms: the primal simplex method, the dual simplex method and the interior-point method. We tested each method individually and selected the shortest computational time as Gurobi's performance benchmark. As illustrated in Fig. 1, the GPU-based sGS-HPR algorithm achieves remarkable computational speedup over Gurobi, particularly for larger-scale systems. For small- and medium-sized instances, Gurobi exhibits higher computational efficiency compared to the sGs-HPR algorithm. One reason for this phenomenon is that the GPU-based sGS-HPR algorithm has a computational overhead of around one second due to the GPU kernel launch time. A turning point occurs when the nnz scale reaches the order of 10^8. Beyond this, as the system size increases further, the efficiency advantage of the proposed sGS-HPR algorithm becomes clearly evident. Overall, the proposed method achieves a 6× speedup compared with Gurobi which operates in its default multi-threaded mode for large-scale instances.

As a performance benchmark against the existing GPU-based algorithm, the speedup ratio of the proposed sGS-HPR algorithm relative to HPR-LP is shown in Fig. 2. It can be observed that the sGS-HPR algorithm also outperforms HPR-LP in computational efficiency. The reasons can be attributed to the following two aspects:

i) The HPR-LP method handles equality and inequality constraints as a whole. Its essence lies in linearizing the entire y-update step, which, although computationally efficient, also leads to an increase in iteration numbers. In contrast, through adopting the sGS decomposition technique, the equality and inequality constraints can be handled separately in the proposed sGS-HPR algorithm. This not only ensures the computational efficiency of the y-update step, but also reduces the total iteration count compared with the HPR-LP method. Although this idea is relatively intuitive, comprehensive theoretical analysis are needed to rigorously establish the convergence and optimality of the proposed sGS-HPR algorithm.

ii) The HPR-LP method is designed for general problems without fully exploiting the problem structure. As a comparison, the proposed sGS-HPR algorithm exploits the structural properties of the DCOPF problems and is specifically designed to take advantage of these characteristics, leading to superior computational efficiency and scalability.

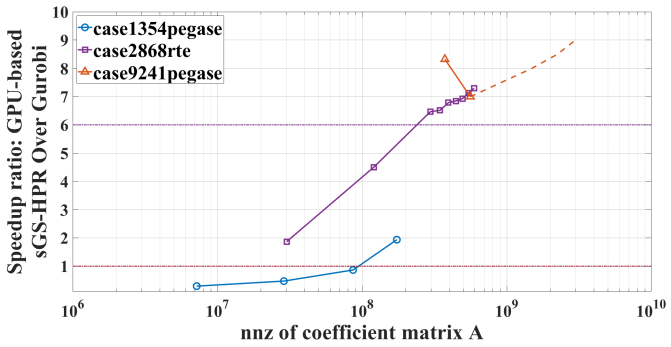

Fig. 1. Speedup ratio of the proposed GPU-based sGS-HPR algorithm over Gurobi.

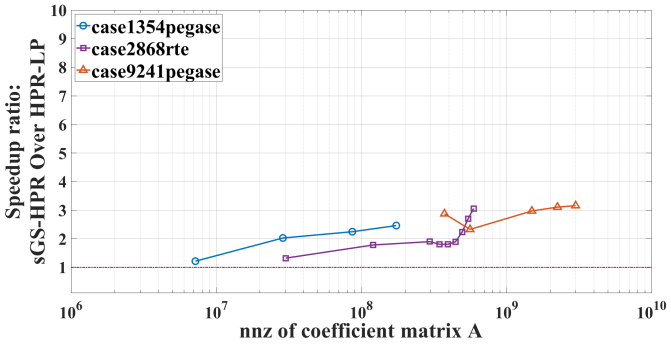

Fig. 2. Speedup ratio of the proposed sGS-HPR algorithm over HPR-LP.

Subsequently, taking the case2868rte system as an example, we compare the computation time growth trends of the sGS-HPR, HPR-LP, cuPDLP and Gurobi, as illustrated in Fig. 3. It can be observed that, as the time periods increase, the growth trend of the proposed GPU-based sGS-HPR algorithm is much smoother compared to the other methods used for benchmarking, further highlighting the advantages of the proposed algorithm in scalability and computational efficiency.

Table II presents the overall numerical results of the proposed sGS-HPR compared to Gurobi, cuPDLP and HPR-LP. The primal simplex method, dual simplex method, and interior-point method embedded in Gurobi are all tested individually. All these three methods encounter an "out-of-memory" error when the nnz scale reaches the order of 10^9. The reason lies in the fact that the simplex method and the interior-point method require LU factorization and Cholesky factorization, respectively. These factorization-based

TABLE II
NUMERICAL RESULTS OF THE PROPOSED SGS-HPR COMPARED TO GUROBI, CUPDLP AND HPR-LP

Cases	Constraint dimension	Variable dimension	nnz of constraint matrix A	Solving time of Gurobi	Solving time of cuPDLP	Solving time of HPR-LP	Solving time of sGS-HPR	Average relative error of obj. between sGS-HPR and Gurobi
case1354 T4	20,192	4,208	7,190,640	1.136	6.194	4.680	3.849	1.573E-05
case1354 T16	82,124	16,832	28,791,792	1.945	11.679	8.338	4.113	4.817E-06
case1354 T48	247,276	50,496	86,586,352	5.848	17.833	15.180	6.757	1.628E-04
case1354 T96	495,004	100,992	173,800,432	15.231	29.671	19.340	**7.857**	1.569E-05
case2868 T4	40,163	9,488	30,111,616	8.489	7.059	6.014	**4.546**	6.331E-05
case2868 T16	163,823	37,952	120,508,576	34.659	17.136	13.717	**7.695**	2.990E-05
case2868 T48	493,583	113,856	295,998,240	90.154	31.787	26.491	**13.948**	1.688E-04
case2868 T56	576,023	132,832	345,459,808	104.152	34.624	28.879	**15.988**	1.076E-04
case2868 T64	658,463	151,808	394,957,984	109.656	38.019	29.209	**16.157**	9.008E-05
case2868 T72	740,903	170,784	444,492,768	115.025	43.451	31.857	**16.808**	4.795E-05
case2868 T80	823,343	189,760	494,064,160	119.797	51.815	38.677	**17.287**	1.019E-04
case2868 T88	905,783	208,736	543,672,160	124.926	62.112	47.456	**17.538**	6.694E-07
case2868 T96	988,223	227,712	593,316,768	130.554	74.872	54.737	**17.899**	4.790E-08
case9241 T4	152,774	24,700	373,238,888	66.904	43.702	23.110	**8.040**	2.635E-05
case9241 T6	230,376	37,050	559,872,262	136.266	87.041	45.333	**19.478**	3.049E-05
case9241 T16	618,386	98,800	1,493,149,532	**OOM**	351.764	177.678	**59.784**	/
case9241 T24	928,794	148,200	2,239,903,828	**OOM**	854.565	430.475	**138.550**	/
case9241 T32	1,239,202	197,600	2,986,775,884	**OOM**	1,492.22	719.247	**227.682**	/

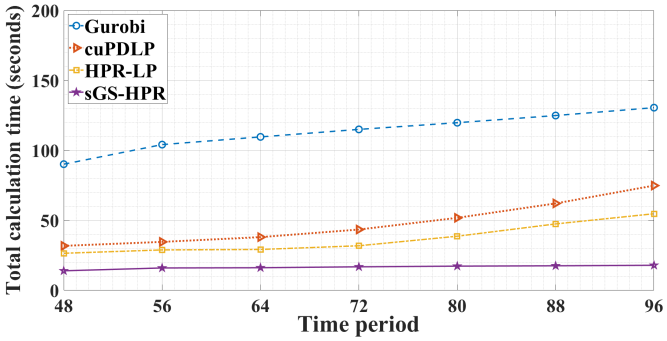

Fig. 3. Comparison of computation time growth trends of the sGS-HPR, HPR-LP, cuPDLP and Gurobi as the time intervals increase.

approaches can be highly memory-consuming. It is often the case that a sparse matrix yields a much denser factorization, which explains why Gurobi raises an "out-of-memory" error even if the problem instance fits within the available memory. In comparison, the proposed GPU-based sGS-HPR algorithm continues to maintain its scalability and efficiency. Overall, the proposed sGS-HPR algorithm exhibits superior performance on large-scale instances compared to Gurobi.

Moreover, the results also demonstrate that the developed sGS-HPR algorithm achieves further performance improvements over the cuPDLP and HPR-LP algorithm. As the problem size increases, the advantages of the proposed algorithm become more apparent. Here, we employ the shifted geometric mean of the solving time for demonstration. The shifted geometric mean is defined as $\left(\prod_{i=1}^{n}(t_i+\Delta)\right)^{1/n}-\Delta$, where t_i denotes the solving time in seconds for the i-th instance. A shift of $\Delta = 10$ is applied, represented as SGM10. Compared with HPR-LP and cuPDLP, the developed sGS-HPR algorithm achieves speedups of 2.12× and 3.05×, respectively. Following the preceding discussion on the advantages of sGS-HPR over HPR-LP, we next conduct a systematic comparison of the update structures of cuPDLP and the developed sGS-HPR algorithm from multiple perspectives, as given in Table III.

In addition, Table II also provides the average relative error of objective values between the proposed sGS-HPR algorithm and Gurobi over five independent runs. It can be observed that the error is sufficiently small, verifying the accuracy and effectiveness of the proposed algorithm.

C. Sensitivity Analysis of Penalty Parameter σ

We compare two scenarios: using a fixed σ and employing the σ update strategy, and further analyze the computational performance for different initial values. For the two cases—one with fixed $\sigma = 1$, and the other with an initial value of 1 along with the σ update strategy, a significant difference in computational efficiency is observed, with the

TABLE III
A COMPARATIVE ANALYSIS OF THE UPDATE STRUCTURES IN CUPDLP AND SGS-HPR

Methods	The form of operator $\mathcal{T}_1$	Relaxation	Halpern iteration	Handling equ. and inequ. cons. separately	Leveraging the DCOPF structure
cuPDLP	$\mathcal{T}_1 = \lambda_1(AA^*)I_m - AA^*$	No	No	No	No
sGS-HPR	$\mathcal{T}_1 = \mathrm{sGS}(\mathcal{H}) + \mathrm{diag}(\mathbf{0}_{m_1}, \mathcal{S}_2)$	Yes	Yes	Yes	Yes

latter achieving a 2.08× speedup when evaluated on SGM10, as given in Table IV. Additionally, the results show that, with the same σ update strategy, varying the initial values has a moderate effect on computation time, as the value of σ adaptively updates in subsequent iterations, diminishing the influence of the initial choice.

TABLE IV
COMPUTATIONAL PERFORMANCE COMPARISON UNDER DIFFERENT INITIAL VALUES AND UPDATE MANNER OF σ.

Cases	SGM10
Fixed $\sigma = 1$	39.037
Initial $\sigma = 1$ with update strategy	18.753
Initial $\sigma = 0.1$ with update strategy	18.591
Initial $\sigma = 10$ with update strategy	18.926

VI. CONCLUSION

With the increasing integration of renewable energy and energy storage systems into power grids, the DCOPF has become a critical tool for power system operation. However, for large-scale power grids, conventional CPU-based optimization algorithms, such as the simplex and barrier methods, have saturated in terms of computational efficiency and memory usage. To address these challenges, this work proposes a GPU-based Halpern Peaceman-Rachford algorithm incorporating the symmetric Gauss–Seidel decomposition. The proposed algorithm exhibits an $O(1/k)$ iteration complexity regarding the KKT residual.

In addition, the closed-form solutions for all subproblems are derived, which involve only matrix-vector multiplications and vector operations, making them highly amenable to GPU parallelization. Consequently, the developed sGS-HPR algorithm achieves a non-ergodic overall computational complexity of $O(N_L \times n/\epsilon)$ for floating-point operations when obtaining an ϵ-optimal solution measured by the KKT residual in large-scale DCOPF problems.

Extensive numerical experiments on multiple power grids, reaching up to the 9241-bus PEGASE system, highlight the scalability and superior efficiency of the developed GPU-based sGS-HPR algorithm in comparison to state-of-the-art methods, including the cuPDLP, HPR-LP and Gurobi v12.0.0, particularly for large-scale problems. Remarkably, the proposed method achieves a 6× speedup over Gurobi operating in its default multi-threaded mode for large-scale instances. Additionally, for ultra-large-scale cases, Gurobi throws an "out-of-memory" error. In contrast, the proposed GPU-based sGS-HPR algorithm maintains its scalability and efficiency, demonstrating its superior performance.

APPENDIX

Appendix A: Proof of Proposition 1

Proof: To facilitate the derivation, the specific form of A_1 is presented in equation (55).

(1) First-row Block

The first-row block consists of:

$$\mathbf{I}_T \otimes \mathbf{1}_{N_{\mathrm{G}}}^*, \quad \mathbf{I}_T \otimes \mathbf{1}_{N_{\mathrm{RG}}}^*, \quad \mathbf{I}_T \otimes \mathbf{1}_{N_{\mathrm{ESS}}}^*, \quad -\mathbf{I}_T \otimes \mathbf{1}_{N_{\mathrm{ESS}}}^*.$$

Utilizing the rank property of the Kronecker product, we have

$$\mathrm{rank}(\mathbf{I}_T \otimes \mathbf{1}_N) = T \cdot \mathrm{rank}(\mathbf{1}_N). \tag{56}$$

Since $\mathbf{1}_N^*$ is an all-one vector of size N, its rank is 1. Thus, the rank of each term in the first-row block is T, and their combination ensures that the first-row block has rank T.

(2) Second-row Block

The second-row block consists of:

$$\mathbf{I}_T \otimes \mathrm{diag}\left(-\frac{\Delta t}{\eta^{\mathrm{dc}}}\right), \quad \mathbf{I}_T \otimes \mathrm{diag}\left(\Delta t \cdot \eta^{\mathrm{ch}}\right).$$

Since $\mathrm{diag}(-\frac{\Delta t}{\eta^{\mathrm{dc}}})$ and $\mathrm{diag}(\Delta t \cdot \eta^{\mathrm{ch}})$ are diagonal matrices with nonzero elements, we have:

$$\begin{aligned} \mathrm{rank}\left(\mathbf{I}_T \otimes \mathrm{diag}\left(-\frac{\Delta t}{\eta^{\mathrm{dc}}}\right)\right) &= T \cdot N_{\mathrm{ESS}}, \\ \mathrm{rank}\left(\mathbf{I}_T \otimes \mathrm{diag}\left(-\frac{\Delta t}{\eta^{\mathrm{dc}}}\right)\right) &= T \cdot N_{\mathrm{ESS}}. \end{aligned} \tag{57}$$

This guarantees that the second-row block has rank $T \cdot N_{\mathrm{ESS}}$.

Equation (55) clearly indicates that the first-row block and second-row block do not have linearly dependent rows. Since A_1 has exactly $T(1 + N_{\mathrm{ESS}})$ rows, it follows that A_1 has full row rank.

This completes the proof.

■

Appendix B: Proof of Proposition 2

Revisit the y-update step in Algorithm 1, it can be reformulated as the following form:

$$\begin{aligned} \bar{y}^{k+1} = \underset{y \in \mathbb{R}^m}{\arg\min} \Bigg\{ & -\langle b, y\rangle + \delta_D(y) + \delta_C^*(-\bar{z}^{k+1}) + \\ & \langle \bar{x}^{k+1}, A^*y + \bar{z}^{k+1} - c\rangle \\ & + \frac{\sigma}{2}\left\|A^*y + \bar{z}^{k+1} - c\right\|^2 + \frac{\sigma}{2}\|y - y^k\|_{\mathcal{T}_1}^2 \Bigg\}, \end{aligned} \tag{58}$$

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{I}_T \otimes \mathbf{1}_{N_{\mathrm{G}}}^{\mathrm{T}} & \mathbf{I}_T \otimes \mathbf{1}_{N_{\mathrm{RG}}}^{\mathrm{T}} & \mathbf{I}_T \otimes \mathbf{1}_{N_{\mathrm{ESS}}}^{\mathrm{T}} & -\mathbf{I}_T \otimes \mathbf{1}_{N_{\mathrm{ESS}}}^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1}_T^{\mathrm{T}} \otimes \mathrm{diag}\left(-\frac{\Delta t}{\eta^{\mathrm{dc}}}\right) & \mathbf{1}_T^{\mathrm{T}} \otimes \mathrm{diag}\left(\Delta t \cdot \eta^{\mathrm{ch}}\right) & \mathbf{0} \end{bmatrix}, \tag{55}$$

where the first-row block represents the power balance constraint. The second-row block indicates that the energy in the first and last periods of the ESS should be equal. $\mathbf{I}_T$ is an identity matrix of dimension T. $\mathbf{1}_N$ is an all-ones column vector of size N. $\mathbf{0}$ represents the zero matrix. $\mathrm{diag}(\cdot)$ stands for diagonal matrix.

which is essentially a quadratic programming problem shown as follows:

$$\begin{aligned} \bar{y}^{k+1} = \underset{y\in\mathbb{R}^m}{\arg\min} \Big\{ & \delta_{\mathbb{R}_+^{m_2}}(y_2) + \frac{1}{2}\langle y, \mathcal{Q}y\rangle - \langle \tilde{b}, y\rangle \\ & + \frac{1}{2}\left\|y - y^k\right\|_{\mathcal{T}_\mathcal{Q}}^2 + \tilde{c} \Big\}, \end{aligned} \tag{59}$$

where

$$\mathcal{Q} = \sigma\mathcal{H}, \quad \mathcal{T}_\mathcal{Q} = \sigma \cdot \mathrm{sGS}(\mathcal{H}). \tag{60}$$

$$\tilde{b} = b - A\bar{x}^{k+1} - \sigma A(\bar{z}^{k+1} - c) + \sigma\mathcal{T}_1 y^k. \tag{61}$$

$$\tilde{c} = \delta_C^*(-\bar{z}^{k+1}) + \langle \bar{x}^{k+1}, \bar{z}^{k+1} - c\rangle + \frac{\sigma}{2}\|\bar{z}^{k+1} - c\|^2 + \frac{\sigma}{2}\|y^k\|_{\mathcal{T}_1}^2. \tag{62}$$

Given that $\tilde{c}$ is a constant, problem (59) can be further rewritten as:

$$\bar{y}^{k+1} = \underset{y\in\mathbb{R}^m}{\arg\min} \left\{ p\left(y_2\right) + q(y) + \frac{1}{2}\|y - y^k\|_{\mathcal{T}_\mathcal{Q}}^2 \right\}, \tag{63}$$

where $p\left(y_2\right) = \delta_{\mathbb{R}_+^{m_2}}(y_2)$, $q(y) = \frac{1}{2}\langle y, \mathcal{Q}y\rangle - \langle \tilde{b}, y\rangle$.

$$\mathcal{Q} = \begin{bmatrix} Q_{1,1} & Q_{1,2} \\ Q_{1,2}^* & Q_{2,2} \end{bmatrix}, \tilde{b} = \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \end{bmatrix},$$

with $Q_{i,j} \in \mathbb{R}^{m_i \times m_j}$ for $1 \le i, j \le 2$.

Assume that $\bar{y}_1^{k+1/2}$ has been computed, as defined by

$$\begin{aligned} \bar{y}_1^{k+1/2} &= \underset{y_1\in\mathbb{R}^{m_1}}{\arg\min} \left\{ p\left(y_2^k\right) + q\left(y_1; y_2^k\right) \right\} \\ &= Q_{1,1}^{-1}\left(\tilde{b}_1 - Q_{1,2}y_2^k\right). \end{aligned} \tag{64}$$

Then the establishment of Proposition 2 is equivalent to proving the following steps yield the optimal solution $\bar{y}^{k+1}$ for (63):

$$\begin{cases} \bar{y}_2^{k+1} = \underset{y_2\in\mathbb{R}^{m_2}}{\arg\min} \left\{ p\left(y_2\right) + q\left(\bar{y}_1^{k+1/2}; y_2\right) \right\}, \\ \bar{y}_1^{k+1} = \underset{y_1\in\mathbb{R}^{m_1}}{\arg\min} \left\{ p\left(\bar{y}_2^{k+1}\right) + q\left(y_1; \bar{y}_2^{k+1}\right) \right\} \\ \qquad\quad = Q_{1,1}^{-1}\left(\tilde{b}_1 - Q_{1,2}\bar{y}_2^{k+1}\right). \end{cases} \tag{65}$$

Below, we establish the equivalence between (63) and (65):

Proof: Since $D_\mathcal{H} \succ 0$, we know that $D_\mathcal{H}, D_\mathcal{H} + U_\mathcal{H}$ and $D_\mathcal{H} + U_\mathcal{H}^*$ are all nonsingular. Then, combining (18)-(21) and (60), we can obtain the positive definiteness of $\mathcal{T}_1 + AA^*$ and $\mathcal{Q} + \mathcal{T}_\mathcal{Q}$:

$$\begin{aligned} AA^* + \mathcal{T}_1 &= (AA^* + \mathrm{sGS}(\mathcal{H}) + \mathrm{diag}(\mathbf{0}_{m_1}, \mathcal{S}_2)) \\ &= (\mathcal{H} + \mathrm{sGS}(\mathcal{H})) \\ &= (D_\mathcal{H} + U_\mathcal{H}^*)D_\mathcal{H}^{-1}(D_\mathcal{H} + U_\mathcal{H}) \succ 0. \end{aligned} \tag{66}$$

Furthermore, given that $\sigma > 0$, it follows that

$$\mathcal{Q} + \mathcal{T}_\mathcal{Q} = \sigma(\mathcal{H} + \mathrm{sGS}(\mathcal{H})) \succ 0. \tag{67}$$

By noting that $\mathcal{Q}_{2,2} \succ 0$, the optimality conditions corresponding to $\bar{y}_2^{k+1}$ in can be expressed as

$$\bar{y}_2^{k+1} = Q_{2,2}^{-1}\left(\tilde{b}_2 - \gamma_2 - Q_{1,2}^*\bar{y}_1^{k+1/2}\right), \tag{68}$$

where $\gamma_2 \in \partial p\left(y_2\right)$.

Formula (68), together with (64), can be equivalently reformulated as:

$$\sigma(D_\mathcal{H} + U_\mathcal{H}^*)y' = \tilde{b} - \gamma - \sigma U_\mathcal{H} y^k, \tag{69}$$

where $y' = \begin{bmatrix} \bar{y}_1^{k+1/2} \\ \bar{y}_2^{k+1} \end{bmatrix}$, $\gamma = \begin{bmatrix} 0 \\ \gamma_2 \end{bmatrix}$.

Similarly, (65) can be equivalently reformulated as:

$$\sigma\left(D_\mathcal{H} + U_\mathcal{H}\right)\bar{y}^{k+1} = \tilde{b} - \gamma - \sigma U_\mathcal{H}^* y'. \tag{70}$$

Based on formula (69), we can obtain that $y' = (D_\mathcal{H} + U_\mathcal{H}^*)^{-1}\left(\frac{\tilde{b}-\gamma}{\sigma} - U_\mathcal{H} y^k\right)$. By substituting it into (70), we have

$$\begin{aligned} & \sigma\left(D_\mathcal{H} + U_\mathcal{H}\right)\bar{y}^{k+1} \\ & = \tilde{b} - \gamma - \sigma U_\mathcal{H}^*(D_\mathcal{H} + U_\mathcal{H}^*)^{-1}\left(\frac{\tilde{b}-\gamma}{\sigma} - U_\mathcal{H} y^k\right) \\ & = D_\mathcal{H}(D_\mathcal{H} + U_\mathcal{H}^*)^{-1}(\tilde{b} - \gamma) + \sigma U_\mathcal{H}^*(D_\mathcal{H} + U_\mathcal{H}^*)^{-1}U_\mathcal{H} y^k. \end{aligned} \tag{71}$$

By multiplying both sides of the above equation by $(D_\mathcal{H} + U_\mathcal{H}^*)D_\mathcal{H}^{-1}$, it follows that:

$$\begin{aligned} & \sigma(D_\mathcal{H} + U_\mathcal{H}^*)D_\mathcal{H}^{-1}\left(D_\mathcal{H} + U_\mathcal{H}\right)\bar{y}^{k+1} \\ & = \tilde{b} - \gamma + \sigma(D_\mathcal{H} + U_\mathcal{H}^*)D_\mathcal{H}^{-1}U_\mathcal{H}^*(D_\mathcal{H} + U_\mathcal{H}^*)^{-1}U_\mathcal{H} y^k. \end{aligned} \tag{72}$$

Then, combined with (67) and the definition of $\mathcal{T}_\mathcal{Q}$ in (60), along with the fact that $(D_\mathcal{H} + U_\mathcal{H}^*)D_\mathcal{H}^{-1}U_\mathcal{H}^*(D_\mathcal{H} + U_\mathcal{H}^*)^{-1} = U_\mathcal{H}^* D_\mathcal{H}^{-1}$, it implies that

$$\left(\mathcal{Q} + \mathcal{T}_\mathcal{Q}\right)\bar{y}^{k+1} = \tilde{b} - \gamma + \mathcal{T}_\mathcal{Q} y^k. \tag{73}$$

Observing that (73) is indeed the optimality condition for (63) and that $\mathcal{Q} + \mathcal{T}_\mathcal{Q} \succ 0$, we thus establish the equivalence between (63) and (65).

This completes the proof.

■

Appendix C: Proof of Proposition 3

Proof: We begin by demonstrating that $(\mathcal{M} + \mathcal{T})^{-1}$ is single-valued through a proof by contradiction. Suppose that $(\mathcal{M} + \mathcal{T})^{-1}$ is not single-valued, then, there would exist two different points $\bar{w}_1 = (\bar{y}_1, \bar{z}_1, \bar{x}_1)$ and $\bar{w}_2 = (\bar{y}_2, \bar{z}_2, \bar{x}_2)$ such that

$$\bar{w}_1, \bar{w}_2 \in (\mathcal{M} + \mathcal{T})^{-1}v \tag{74}$$

for some $v = (v_y, v_z, v_x) \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n$. This implies that for $i = 1, 2$ the following conditions hold:

$$\begin{aligned} v_y &= -b + \sigma\left(AA^* + \mathcal{T}_1\right)\bar{y}_i + 2A\bar{x}_i, \\ v_z &\in -\partial\delta_C^*(-\bar{z}_i) + \bar{x}_i, \\ v_x &= c - \bar{z}_i + \sigma^{-1}\bar{x}_i. \end{aligned} \tag{75}$$

Following this, the following formulas can be derived:

$$\sigma\left(AA^* + \mathcal{T}_1\right)(\bar{y}_1 - \bar{y}_2) + 2A\left(\bar{x}_1 - \bar{x}_2\right) = 0. \tag{76}$$

$$\left\langle -\left(\bar{x}_1 - \bar{x}_2\right), \bar{z}_1 - \bar{z}_2 \right\rangle \geq 0. \tag{77}$$

$$\bar{x}_1 - \bar{x}_2 = \sigma(\bar{z}_1 - \bar{z}_2). \tag{78}$$

Through substituting (78) into (77), we can obtain

$$\bar{z}_1 - \bar{z}_2 = 0, \tag{79}$$

which together with (78) leads to that

$$\bar{x}_1 - \bar{x}_2 = 0. \tag{80}$$

Similarly, by substituting (80) into (76), we can deduce from the positive definiteness of $\sigma\left(AA^* + \mathcal{T}_1\right)$ that

$$\bar{y}_1 - \bar{y}_2 = 0. \tag{81}$$

Thus, equations (79), (80), and (81) imply that $\bar{w}_1 = \bar{w}_2$, which contradicts our assumption. Therefore, $(\mathcal{M} + \mathcal{T})^{-1}$ is single-valued.

Next, we proceed to establish the Lipschitz continuity of $(\mathcal{M} + \mathcal{T})^{-1}$. Consider $\bar{w}_i = (\bar{y}_i, \bar{z}_i, \bar{x}_i) \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n$ such that $\bar{w}_i = (\mathcal{M} + \mathcal{T})^{-1} v_i$, where $v_i = (v_{iy}, v_{iz}, v_{ix})$ for $i = 1, 2$. Similar to (75), we have

$$\begin{aligned} v_{iy} - 2A\bar{x}_i &= -b + \sigma(AA^* + \mathcal{T}_1)\bar{y}_i, \\ v_{iz} - \sigma\left(v_{ix} - c\right) &\in -\partial\delta_C^*(-\bar{z}_i) + \sigma\bar{z}_i, \\ v_{ix} &= c - \bar{z}_i + \sigma^{-1}\bar{x}_i. \end{aligned} \tag{82}$$

Using the findings in [60], we can obtain

$$\begin{aligned} \bar{y}_i &= \left(\sigma(AA^* + \mathcal{T}_1)\right)^{-1}\left(v_{iy} - 2A\bar{x}_i + b\right), \\ \bar{z}_i &\in \left(\partial\delta_C^* + \sigma I_n\right)^{-1}\left(v_{iz} - \sigma\left(v_{ix} - c\right)\right). \end{aligned} \tag{83}$$

Since $\left(\partial\delta_C^* + \sigma I_n\right)^{-1}$ satisfies the non-expansiveness property [56], it follows that there exists a constant L_2 such that

$$\|\bar{z}_1 - \bar{z}_2\| \leq L_2\|v_{1z} - v_{2z}\| + \sigma L_2\|v_{1x} - v_{2x}\|. \tag{84}$$

Furthermore, combined with (82) and (84), we have

$$\begin{aligned} \|\bar{x}_1 - \bar{x}_2\| &= \|\sigma\left(v_{1x} - v_{2x} + \bar{z}_1 - \bar{z}_2\right)\| \\ &\leq \sigma\left(1 + L_2\right)\|v_{1x} - v_{2x}\| + \sigma L_2\|v_{1z} - v_{2z}\|. \end{aligned} \tag{85}$$

Since $\sigma(AA^* + \mathcal{T}_1)$ is strongly monotone, a constant L_1 exists such that

$$\begin{aligned} \|\bar{y}_1 - \bar{y}_2\| \leq& L_1\|v_{1y} - v_{2y} - 2A\left(\bar{x}_1 - \bar{x}_2\right)\| \\ \leq& L_1\|v_{1y} - v_{2y}\| + 2L_1\|A\|\sigma\left(1 + L_2\right)\|v_{1x} - v_{2x}\| \\ &+ 2L_1\|A\|\sigma L_2\|v_{1z} - v_{2z}\|. \end{aligned} \tag{86}$$

Thus, combining (84), (85), and (86), there exists a constant L such that

$$\|\bar{w}_1 - \bar{w}_2\| \leq L\|v_1 - v_2\|, \tag{87}$$

which establishes the Lipschitz continuity of $(\mathcal{M} + \mathcal{T})^{-1}$.

Finally, we demonstrate the equivalence between the sGS-HPR algorithm and the accelerated degenerate proximal point method. Combining (23) and $\sigma > 0$, it follows that $\sigma(AA^* + \mathcal{T}_1)$ and $\sigma\mathcal{I}$ are both positive definite, meaning that they are maximal and strongly monotone.

This further implies that the objective function of each subproblem in Algorithm 2 is a proper, closed, strongly convex function [56].

Hence, it follows that $\bar{z}^{k+1} = \underset{z\in\mathbb{R}^n}{\arg\min}\left\{L_\sigma\left(y_1^k, y_2^k, z; x^k\right)\right\}$ holds if and only if the following equation holds:

$$0 \in -\partial\delta_C^*(-\bar{z}^{k+1}) + x^k + \sigma\left(A^*y^k + \bar{z}^{k+1} - c\right). \tag{88}$$

Similarly, equation $\bar{y}^{k+1} = \underset{y\in\mathbb{R}^m}{\arg\min}\left\{L_\sigma\left(y, \bar{z}^{k+1}; \bar{x}^{k+1}\right) + \frac{\sigma}{2}\|y - y^k\|_{\mathcal{T}_1}^2\right\}$ holds if and only if the following equation is satisfied:

$$\begin{aligned} 0 \in -b + \mathcal{N}_D(\bar{y}^{k+1}) + A\left(\bar{x}^{k+1} + \sigma\left(A^*\bar{y}^{k+1} + \bar{z}^{k+1} - c\right)\right) \\ + \sigma\mathcal{T}_1\left(\bar{y}^{k+1} - y^k\right). \end{aligned} \tag{89}$$

Therefore, the update steps in Algorithm 2 can be equivalently reformulated as:

$$\begin{cases} 0 \in -\partial\delta_C^*(-\bar{z}^{k+1}) + x^k + \sigma\left(A^*y^k + \bar{z}^{k+1} - c\right), \\ \bar{x}^{k+1} = x^k + \sigma\left(A^*y^k + \bar{z}^{k+1} - c\right), \\ 0 \in -b + \mathcal{N}_D(\bar{y}^{k+1}) + A\left(\bar{x}^{k+1} + \sigma\left(A^*\bar{y}^{k+1} + \bar{z}^{k+1} - c\right)\right) \\ \quad + \sigma\mathcal{T}_1\left(\bar{y}^{k+1} - y^k\right), \\ \hat{w}^{k+1} = 2\bar{w}^{k+1} - w^k, \\ w^{k+1} = \dfrac{1}{k+2}w^0 + \dfrac{k+1}{k+2}\hat{w}^{k+1}. \end{cases} \tag{90}$$

Notice that for any $\left(y^{k+1}, z^{k+1}, x^{k+1}\right)$ satisfying (90), the following holds:

$$\begin{aligned} -\sigma^{-1}\left(\bar{x}^{k+1} - x^k\right) &= -\left(A^*y^k + \bar{z}^{k+1} - c\right) \\ &= c - \bar{z}^{k+1} - A^*\bar{y}^{k+1} + A^*\left(\bar{y}^{k+1} - y^k\right). \end{aligned} \tag{91}$$

In this way, (90) can be rewritten as

$$\begin{cases} 0 \in -\partial\delta_C^*(-\bar{z}^{k+1}) + \bar{x}^{k+1}, \\ \bar{x}^{k+1} = x^k + \sigma\left(A^*y^k + \bar{z}^{k+1} - c\right), \\ 0 \in -b + \mathcal{N}_D(\bar{y}^{k+1}) + A\bar{x}^{k+1} + \sigma\left(AA^* + \mathcal{T}_1\right)\left(\bar{y}^{k+1} - y^k\right) \\ \quad + A\left(\bar{x}^{k+1} - x^k\right), \\ \hat{w}^{k+1} = 2\bar{w}^{k+1} - w^k, \\ w^{k+1} = \dfrac{1}{k+2}w^0 + \dfrac{k+1}{k+2}\hat{w}^{k+1}. \end{cases} \tag{92}$$

Combined with the definition of $\mathcal{T}$ and $\mathcal{M}$, (92) can be further reformulated as:

$$\begin{cases} \mathcal{M}w^k \in (\mathcal{M} + \mathcal{T})\bar{w}^{k+1}, \\ \hat{w}^{k+1} = 2\bar{w}^{k+1} - w^k, \\ w^{k+1} = \dfrac{1}{k+2}w^0 + \dfrac{k+1}{k+2}\hat{w}^{k+1}. \end{cases} \tag{93}$$

Since $(\mathcal{M}+\mathcal{T})^{-1}$ is single-valued, (93) can be rewritten as

$$\begin{cases} \bar{w}^{k+1} = (\mathcal{M}+\mathcal{T})^{-1}\mathcal{M}w^k, \\ \hat{w}^{k+1} = 2\bar{w}^{k+1} - w^k, \\ w^{k+1} = \dfrac{1}{k+2}w^0 + \dfrac{k+1}{k+2}\hat{w}^{k+1}. \end{cases} \tag{94}$$

which is exactly the scheme (27).

Hence, the point $\left(y^{k+1}, z^{k+1}, x^{k+1}\right)$ generated by Algorithm 2 is identical to the one generated by the accelerated dPPM (27).

Given an arbitrary choice of $w^k \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n$, each step in Algorithm 2 is well-defined. Therefore, based on the derived equivalence, it follows that $(\mathcal{M}+\mathcal{T})^{-1}\mathcal{M}$ has a full domain. By combining this with the Lipschitz continuity result derived earlier, it can be concluded that $\mathcal{M}$ is an admissible preconditioner such that $(\mathcal{M}+\mathcal{T})^{-1}$ is Lipschitz continuous.

This completes the proof.

■

Appendix D: Proof of Proposition 4

Given the operator $\mathcal{M}$ defined in (26), there exists a bounded and injective operator $\mathcal{C}$ such that $\mathcal{M} = \mathcal{C}\mathcal{C}^*$, where $\mathcal{C}^*$ denotes the adjoint of $\mathcal{C}$ [42]. Along with the operator $\mathcal{T}$ defined in (25), we then define

$$\begin{aligned} \widetilde{\mathcal{T}} &= \mathcal{C}^*(\mathcal{M}+\mathcal{T})^{-1}\mathcal{C}, \\ \widetilde{\mathcal{F}} &= 2\widetilde{\mathcal{T}} - \mathcal{I}, \end{aligned} \tag{95}$$

where $\mathcal{I}$ represents an identity matrix.

The following proposition provides a summary of some fundamental properties of $\widetilde{\mathcal{T}}$ and $\widetilde{\mathcal{F}}$.

Proposition 6. *(Proposition 2.5 in [37]) Given the operators $\mathcal{T}$ defined in (25) and $\mathcal{M}$ defined in (26), it follows that the operator $\widetilde{\mathcal{T}}$ defined in (95) is well-defined and firmly-nonexpansive. Moreover, the operator $\widetilde{\mathcal{F}}$ in (95) is non-expansive, and we further have*

$$\mathcal{C}^*\mathcal{T}^{-1}(0) = \mathcal{C}^* \operatorname{Fix}\widehat{\mathcal{T}} = \operatorname{Fix}\widetilde{\mathcal{T}} = \operatorname{Fix}\widetilde{\mathcal{F}}, \tag{96}$$

where the set of fixed points of an operator $\widehat{\mathcal{T}}$ is denoted as Fix $\widehat{\mathcal{T}}$.

For the purpose of analyzing the global convergence and iteration complexity of the Algorithm 2, we define two auxiliary sequences $\left\{u^k\right\}$ and $\left\{\bar{u}^k\right\}$ as follows:

$$u^k := \mathcal{C}^*w^k \text{ and } \bar{u}^k := \mathcal{C}^*\bar{w}^k, \quad \forall k \geq 0, \tag{97}$$

where the sequences $\left\{w^k\right\}$ and $\left\{\bar{w}^k\right\}$ are generated by Algorithm 2.

Utilizing Proposition 3, we can derive the following equation:

$$u^{k+1} = \frac{1}{k+2}u^0 + \frac{k+1}{k+2}\widetilde{\mathcal{F}}u^k, \quad \forall k \geq 0. \tag{98}$$

We then proceed with the proof of Proposition 4.

Proof of Proposition 4: It can be observed that the scheme in (98) is the Halpern iteration applied to the non-expansive operator $\widetilde{\mathcal{F}}$. Based on the global convergence of the Halpern iteration established in [61, Theorem 2], it follows that

$$u^k \to u^*, \tag{99}$$

where u^* is a point in $\operatorname{Fix}\widetilde{\mathcal{F}}$.

Employing Proposition 6, we have

$$\mathcal{C}^*\mathcal{T}^{-1}(0) = \mathcal{C}^* \operatorname{Fix}\widehat{\mathcal{T}} = \operatorname{Fix}\widetilde{\mathcal{T}}, \tag{100}$$

which deduces that there exists a w^* in $\mathcal{T}^{-1}(0)$ such that $\mathcal{C}^*w^* = u^*$.

Utilizing the relationship between $\left\{u^k\right\}$ and $\left\{w^k\right\}$ in (97), and (99), we can obtain

$$\begin{aligned} &\bar{w}^k = (\mathcal{M}+\mathcal{T})^{-1}\mathcal{C}C^*w^k = (\mathcal{M}+\mathcal{T})^{-1}\mathcal{C}u^k \\ &\to (\mathcal{M}+\mathcal{T})^{-1}\mathcal{C}\mathcal{C}^*w^* = (\mathcal{M}+\mathcal{T})^{-1}\mathcal{M}w^* = w^*, \end{aligned} \tag{101}$$

where the continuity of $(\mathcal{M}+\mathcal{T})^{-1}\mathcal{C}$ follows from the composition of a continuous function $(\mathcal{M}+\mathcal{T})^{-1}$ demonstrated in Proposition 3 and a linear operator $\mathcal{C}$.

Therefore, $\left\{\bar{w}^k\right\}$ converges to w^*, which completes the proof.

■

Appendix E: Proof of Theorem 1

Proof: It can be observed that the shadow sequence $\left\{u^k\right\}$, as defined in (98) corresponds exactly to the Halpern iteration. According to Proposition 6, it follows that $\widetilde{\mathcal{F}}$ is non-expansive. From [39, Theorem 2.1], we can deduce that

$$\left\|u^k - \widetilde{\mathcal{F}}u^k\right\| \leq \frac{2\left\|u^0 - u^*\right\|}{k+1}, \quad \forall k \geq 0 \text{ and } u^* \in \operatorname{Fix}\widetilde{\mathcal{F}}. \tag{102}$$

By applying Proposition 6, we can also obtain $\operatorname{Fix}\widetilde{\mathcal{F}} = \mathcal{C}^*\mathcal{T}^{-1}(0)$. Therefore, for any $u^* \in \operatorname{Fix}\widetilde{\mathcal{F}}$, there exists a point $w^* = (\mathcal{M}+\mathcal{T})^{-1}\mathcal{C}u^* \in \mathcal{T}^{-1}(0)$ such that $\mathcal{C}^*w^* = u^*$.

Consequently, (102) can be rewritten as:

$$\begin{aligned} &\left\|\mathcal{C}^*w^k - \mathcal{C}^*\widehat{\mathcal{F}}w^k\right\| \leq \frac{2\left\|\mathcal{C}^*w^0 - \mathcal{C}^*w^*\right\|}{k+1}, \quad \forall k \geq 0 \\ &\qquad \text{and } w^* \in \mathcal{T}^{-1}(0), \end{aligned} \tag{103}$$

which implies

$$\begin{aligned} \left\|w^k - \hat{w}^{k+1}\right\|_{\mathcal{M}} &= \left\|w^k - \widehat{\mathcal{F}}w^k\right\|_{\mathcal{M}} \\ &\leq \frac{2\left\|w^0 - w^*\right\|_{\mathcal{M}}}{k+1}, \quad \forall k \geq 0 \text{ and } w^* \in \mathcal{T}^{-1}(0). \end{aligned} \tag{104}$$

Next, we estimate the convergence rate of $\mathcal{R}\left(\bar{w}^k\right)$ for any $k \geq 0$. Based on (104), we have

$$\left\|\hat{w}^{k+1} - w^k\right\|_{\mathcal{M}}^2 \leq \frac{4R_0^2}{(k+1)^2}, \quad \forall k \geq 0. \tag{105}$$

Adopting the definition of $\mathcal{M}$ in (26), inequality (105) can be reformulated as

$$\begin{aligned} &\sigma\left\|\hat{y}^{k+1} - y^k\right\|_{\mathcal{T}_1}^2 + \frac{1}{\sigma}\left\|\sigma A^*\left(\hat{y}^{k+1} - y^k\right) + \left(\hat{x}^{k+1} - x^k\right)\right\|^2 \\ &\qquad \leq \frac{4R_0^2}{(k+1)^2}, \forall k \geq 0. \end{aligned} \tag{106}$$

Moreover, from Step 4 in Algorithm 2, we can obtain that for any $k \geq 0$,

$$\begin{cases} \hat{y}^{k+1} - y^k = 2\left(\bar{y}^{k+1} - y^k\right), \\ \hat{z}^{k+1} - z^k = 2\left(\bar{z}^{k+1} - z^k\right), \\ \hat{x}^{k+1} - x^k = 2\left(\bar{x}^{k+1} - x^k\right). \end{cases} \tag{107}$$

Therefore, (106) can be rewritten as

$$\sigma\left\|\bar{y}^{k+1} - y^k\right\|_{\mathcal{T}_1}^2 + \frac{1}{\sigma}\left\|\sigma A^*\left(\bar{y}^{k+1} - y^k\right) + \left(\bar{x}^{k+1} - x^k\right)\right\|^2 \leq \frac{R_0^2}{(k+1)^2}, \forall k \geq 0. \tag{108}$$

Given Step 2 in Algorithm 2, we can assert that for any $k \geq 0$,

$$\begin{aligned} &\left\|\sigma A^*\left(\bar{y}^{k+1} - y^k\right) + \left(\bar{x}^{k+1} - x^k\right)\right\| \\ &= \left\|\sigma A^*\left(\bar{y}^{k+1} - y^k\right) + \sigma\left(A^* y^k + \bar{z}^{k+1} - c\right)\right\| \\ &= \sigma\left\|A^*\bar{y}^{k+1} + \bar{z}^{k+1} - c\right\|, \end{aligned} \tag{109}$$

which in conjunction with (108) implies that

$$\begin{aligned} \left\|A^*\bar{y}^{k+1} + \bar{z}^{k+1} - c\right\| &\leq \frac{R_0}{\sqrt{\sigma}(k+1)}, \quad \forall k \geq 0, \\ \left\|\bar{y}^{k+1} - y^k\right\|_{\mathcal{T}_1} &\leq \frac{R_0}{\sqrt{\sigma}(k+1)}, \quad \forall k \geq 0. \end{aligned} \tag{110}$$

Furthermore, based on the optimality conditions of the subproblems in Algorithm 2, for any $k \geq 0$, we have

$$\begin{aligned} \bar{y}^{k+1} &= \mathrm{Prox}_{f_1}\Big(\bar{y}^{k+1} - A\left(\bar{x}^{k+1} + \sigma\left(A^*\bar{y}^{k+1} + \bar{z}^{k+1} - c\right)\right) \\ &\quad - \sigma\mathcal{T}_1\left(\bar{y}^{k+1} - y^k\right)\Big), \\ \bar{z}^{k+1} &= \mathrm{Prox}_{f_2}\left(\bar{z}^{k+1} - \bar{x}^{k+1}\right), \end{aligned} \tag{111}$$

where $f_1 = -\langle b, y\rangle + \delta_D(y)$, $f_2 = \delta_C^*(-z)$.

From (109)-(111), for any $k \geq 0$, we can also obtain

$$\begin{aligned} &\left\|\bar{y}^{k+1} - \mathrm{Prox}_{f_1}\left(\bar{y}^{k+1} - A\bar{x}^{k+1}\right)\right\| \\ \leq &\left\|\sigma A\left(A^*\bar{y}^{k+1} + \bar{z}^{k+1} - c\right) + \sigma\mathcal{T}_1\left(\bar{y}^{k+1} - y^k\right)\right\| \\ \leq &\sigma\|A\|\left\|A^*\bar{y}^{k+1} + \bar{z}^{k+1} - c\right\| + \sigma\left\|\mathcal{T}_1\left(\bar{y}^{k+1} - y^k\right)\right\| \\ \leq &\sigma\|A\|\left\|A^*\bar{y}^{k+1} + \bar{z}^{k+1} - c\right\| + \sigma\sqrt{\mathcal{T}_1}\left\|\bar{y}^{k+1} - y^k\right\|_{\mathcal{T}_1} \\ \leq &\sqrt{\sigma}\left(\|A\| + \left\|\sqrt{\mathcal{T}_1}\right\|\right)\frac{R_0}{(k+1)}. \end{aligned} \tag{112}$$

In addition, $\mathcal{R}(w)$ as defined in (28) can be equivalently expressed as follows:

$$\begin{aligned} \mathcal{R}(w) &= \begin{pmatrix} y - \mathrm{Prox}_{f_1}(y - Ax) \\ z - \mathrm{Prox}_{f_2}(z - x) \\ c - A^*y - z \end{pmatrix}, \\ \forall w = (y, z, x) &\in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n. \end{aligned} \tag{113}$$

As a result, by combining (110)-(112), we can derive that for any $k \geq 0$,

$$\begin{aligned} \left\|\mathcal{R}\left(\bar{w}^{k+1}\right)\right\| &\leq \sqrt{\sigma}\left(\|A\| + \left\|\sqrt{\mathcal{T}_1}\right\|\right)\frac{R_0}{(k+1)} + \frac{R_0}{\sqrt{\sigma}(k+1)} \\ &= \left(\frac{\sigma\left(\|A^*\| + \left\|\sqrt{\mathcal{T}_1}\right\|\right) + 1}{\sqrt{\sigma}}\right)\frac{R_0}{(k+1)}. \end{aligned} \tag{114}$$

This completes the proof. ■

REFERENCES

[1] H. Cheng, X. Liao, H. Li, and Q. Lü, "Dynamic-based privacy preservation for distributed economic dispatch of microgrids," *IEEE Transactions on Control of Network Systems*, pp. 1–11, 2024.

[2] L. Fan, C. Zhao, G. Zhang, and Q. Huang, "Flexibility management in economic dispatch with dynamic automatic generation control," *IEEE Transactions on Power Systems*, vol. 37, no. 2, pp. 876–886, 2022.

[3] Z. Shao, W. Wang, B. Eldridge, A. Somani, and L. Wu, "Real-time economic dispatch approach for wholesale energy market with multi-transmission-node DER aggregation," *IEEE Transactions on Power Systems*, pp. 1–12, 2024.

[4] M. Chávez-Lugo, C. R. Fuerte-Esquivel, C. A. Cañizares, and V. J. Gutierrez-Martinez, "Practical security boundary-constrained DC optimal power flow for electricity markets," *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 3358–3368, 2016.

[5] T. Xu, C. Shao, M. Shahidehpour, and X. Wang, "Coordinated planning strategies of power systems and energy transportation networks for resilience enhancement," *IEEE Transactions on Sustainable Energy*, vol. 14, no. 2, pp. 1217–1229, 2023.

[6] O. M. Adeyanju, P. Siano, and L. N. Canha, "Dedicated microgrid planning and operation approach for distribution network support with pumped-hydro storage," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 7, pp. 8229–8241, 2023.

[7] A. V. Ramesh, X. Li, and K. W. Hedman, "An accelerated-decomposition approach for security-constrained unit commitment with corrective network reconfiguration," *IEEE Transactions on Power Systems*, vol. 37, no. 2, pp. 887–900, 2022.

[8] Y. Chen, *et al.*, "Security-constrained unit commitment for electricity market: Modeling, solution methods, and future challenges," *IEEE Transactions on Power Systems*, vol. 38, no. 5, pp. 4668–4681, 2023.

[9] H. Husin, M. Zaki, *et al.*, "A critical review of the integration of renewable energy sources with various technologies," *Protection and control of modern power systems*, vol. 6, no. 1, pp. 1–18, 2021.

[10] J. Guo, Y. Jing, W. Hou, T. Wang, S. Ma, and G. He, "Demands and challenges of energy storage technology for future power system," *Energy Internet*, vol. 1, no. 2, pp. 116–122, 2024.

[11] F. Ahsan, *et al.*, "Data-driven next-generation smart grid towards sustainable energy evolution: techniques and technology review," *Protection and Control of Modern Power Systems*, vol. 8, no. 3, pp. 1–42, 2023.

[12] A. Kargarian, J. Mohammadi, J. Guo, S. Chakrabarti, M. Barati, G. Hug, S. Kar, and R. Baldick, "Toward distributed/decentralized DC optimal power flow implementation in future electric power systems," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2574–2594, 2018.

[13] M. Alkhraijah, M. Alowaifeer, S. Grijalva, and D. K. Molzahn, "Distributed multi-period DCOPF via an auxiliary principle problem algorithm," in *2021 IEEE Texas Power and Energy Conference (TPEC)*, 2021, pp. 1–6.

[14] A. Bagheri and S. Jadid, "A robust distributed market-clearing model for multi-area power systems," *International Journal of Electrical Power & Energy Systems*, vol. 124, p. 106275, 2021.

[15] Q. Wang, W. Wu, C. Lin, Y. Yang, and B. Wang, "A spatio-temporal decomposition method for the coordinated economic dispatch of integrated transmission and distribution grids," *IEEE Transactions on Power Systems*, vol. 39, no. 3, pp. 4835–4851, 2024.

[16] P. Braun, L. Grüne, C. M. Kellett, S. R. Weller, and K. Worthmann, "A distributed optimization algorithm for the predictive control of smart grids," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3898–3911, 2016.

[17] Gurobi Optimization, "Gurobi optimizer reference manual," 2025. [Online]. Available: https://www.gurobi.com

[18] IBM, "User's manual for cplex," 2025. [Online]. Available: https://www.ibm.com/products/ilog-cplex-optimization-studio

[19] H. Lu and J. Yang, "cuPDLP. jl: A GPU implementation of restarted primal-dual hybrid gradient for linear programming in Julia," *Operations Research*, 2025.

[20] ——, "An overview of GPU-based first-order methods for linear programming and extensions," *arXiv preprint arXiv:2506.02174*, 2025.

[21] A. Degleris, A. E. Gamal, and R. Rajagopal, "GPU accelerated security constrained optimal power flow," *arXiv:2410.17203*, 2024.

[22] S. Saba Rafiei and S. Chevalier, "GPU-Accelerated dcopf using gradient-based optimization," *arXiv:2406.13191*, 2024.

[23] X. Zeng, J. Lei, and J. Chen, "Dynamical primal-dual nesterov accelerated method and its application to network optimization," *IEEE Transactions on Automatic Control*, vol. 68, no. 3, pp. 1760–1767, 2022.
[24] N. Bastianello, R. Carli, L. Schenato, and M. Todescato, "Asynchronous distributed optimization over lossy networks via relaxed ADMM: Stability and linear convergence," *IEEE Transactions on Automatic Control*, vol. 66, no. 6, pp. 2620–2635, 2020.
[25] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.
[26] M. Schubiger, G. Banjac, and J. Lygeros, "GPU acceleration of ADMM for large-scale quadratic programming," *Journal of Parallel and Distributed Computing*, vol. 144, pp. 55–67, 2020.
[27] D. Davis and W. Yin, "Convergence rate analysis of several splitting schemes," in *Splitting Methods in Communication, Imaging, Science, and Engineering*, R. Glowinski, S. J. Osher, and W. Yin, Eds. Springer, 2016, pp. 115–163.
[28] Y. Cui, X. Li, D. Sun, and K.-C. Toh, "On the convergence properties of a majorized alternating direction method of multipliers for linearly constrained convex optimization problems with coupled objective functions," *Journal of Optimization Theory and Applications*, vol. 169, pp. 1013–1041, 2016.
[29] D. Applegate, M. Díaz, O. Hinder, H. Lu, M. Lubin, B. O'Donoghue, and W. Schudy, "Practical large-scale linear programming using primal-dual hybrid gradient," *Advances in Neural Information Processing Systems*, vol. 34, pp. 20 243–20 257, 2021.
[30] D. Applegate, O. Hinder, H. Lu, and M. Lubin, "Faster first-order primal-dual methods for linear programming using restarts and sharpness," *Mathematical Programming*, vol. 201, no. 1, pp. 133–184, 2023.
[31] M. Zhu and T. Chan, "An efficient primal-dual hybrid gradient algorithm for total variation image restoration," *Ucla Cam Report*, vol. 34, no. 2, 2008.
[32] M. Fazel, T. K. Pong, D. F. Sun, and P. Tseng, "Hankel matrix rank minimization with applications to system identification and realization," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 3, pp. 946–977, 2013.
[33] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of mathematical imaging and vision*, vol. 40, pp. 120–145, 2011.
[34] E. Esser, X. Zhang, and T. F. Chan, "A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science," *SIAM Journal on Imaging Sciences*, vol. 3, no. 4, pp. 1015–1046, 2010.
[35] G. Zhang, Y. Yuan, and D. F. Sun, "An efficient HPR algorithm for the Wasserstein barycenter problem with $O(\mathrm{Dim(P)}/\varepsilon)$ computational complexity," *arXiv preprint arXiv:2211.14881*, 2022.
[36] B. Yang, X. Zhao, X. Li, and D. Sun, "An accelerated proximal alternating direction method of multipliers for optimal decentralized control of uncertain systems," *Journal of Optimization Theory and Applications*, vol. 204, no. 1, p. 9, 2025.
[37] D. Sun, Y. Yuan, G. Zhang, and X. Zhao, "Accelerating preconditioned ADMM via degenerate proximal point mappings," *SIAM Journal on Optimization*, vol. 35, no. 2, pp. 1165–1193, 2025.
[38] B. Halpern, "Fixed points of nonexpanding maps," *Bulletin of the American Mathematical Society*, vol. 73, no. 6, pp. 957–961, 1967.
[39] F. Lieder, "On the convergence rate of the Halpern-iteration," *Optimization Letters*, vol. 15, no. 2, pp. 405–418, 2021.
[40] P.-L. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators," *SIAM Journal on Numerical Analysis*, vol. 16, no. 6, pp. 964–979, 1979.
[41] J. Eckstein and D. P. Bertsekas, "On the douglas—rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical programming*, vol. 55, pp. 293–318, 1992.
[42] K. Bredies, E. Chenchene, D. A. Lorenz, and E. Naldi, "Degenerate preconditioned proximal point algorithms," *SIAM Journal on Optimization*, vol. 32, no. 3, pp. 2376–2401, 2022.
[43] K. Chen, D. Sun, Y. Yuan, G. Zhang, and X. Zhao, "HPR-LP: An implementation of an HPR method for solving linear programming," *Mathematical Programming Computation*, pp. 1–28, 2025.
[44] Y. Yang, W. Wu, B. Wang, and M. Li, "Analytical reformulation for stochastic unit commitment considering wind power uncertainty with gaussian mixture model," *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 2769–2782, 2020.
[45] L. Liu, Z. Hu, X. Duan, and N. Pathak, "Data-driven distributionally robust optimization for real-time economic dispatch considering secondary frequency regulation cost," *IEEE Transactions on Power Systems*, vol. 36, no. 5, pp. 4172–4184, 2021.
[46] A. W. Bizuayehu, A. A. Sánchez de la Nieta, J. Contreras, and J. P. S. Catalão, "Impacts of stochastic wind power and storage participation on economic dispatch in distribution systems," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 3, pp. 1336–1345, 2016.
[47] Y. Xia, K. Wang, Y. Huang, T. Lin, L. Shi, and F. Wu, "Bounded rational decision-making modeling and analysis in local energy markets: A state-of-the-art review," *Renewable and Sustainable Energy Reviews*, vol. 226, p. 116310, 2026.
[48] Y. Tang, Q. Zhai, and J. Zhao, "Multi-stage robust economic dispatch with virtual energy storage and renewables based on a single level model," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 4, pp. 5490–5502, 2024.
[49] J. Zhu, K. Zeng, J. Chen, W. Zhao, W. Liu, and W. Zhu, "Transfer-based approximate dynamic programming for rolling security-constrained unit commitment with uncertainties," *Protection and Control of Modern Power Systems*, vol. 9, no. 5, pp. 42–53, 2024.
[50] Q. Wang, C. Lin, W. Wu, B. Wang, G. Wang, H. Liu, H. Zhang, and J. Zhang, "A nested decomposition method for the AC optimal power flow of hierarchical electrical power grids," *IEEE Transactions on Power Systems*, vol. 38, no. 3, pp. 2594–2609, 2022.
[51] W. Wu, J. Chen, B. Zhang, and H. Sun, "A robust wind power optimization method for look-ahead power dispatch," *IEEE Transactions on Sustainable Energy*, vol. 5, no. 2, pp. 507–515, 2014.
[52] Q. Li and V. Vittal, "Non-iterative enhanced SDP relaxations for optimal scheduling of distributed energy storage in distribution systems," *IEEE Transactions on Power Systems*, vol. 32, no. 3, pp. 1721–1732, 2017.
[53] Q. Wang, W. Wu, C. Lin, S. Xu, S. Wang, and J. Tian, "An exact relaxation method for complementarity constraints of energy storages in power grid optimization problems," *Applied Energy*, vol. 371, p. 123592, 2024.
[54] X. Li, D. F. Sun, and K.-C. Toh, "A Schur complement based semi-proximal ADMM for convex quadratic conic programming and extensions," *Mathematical Programming*, vol. 155, no. 1-2, pp. 333–373, 2016.
[55] X. Li, D. Sun, and K.-C. Toh, "A block symmetric Gauss–Seidel decomposition theorem for convex composite quadratic programming and its applications," *Mathematical Programming*, vol. 175, pp. 395–418, 2019.
[56] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*. Springer, New York, 1998.
[57] D. Han, D. F. Sun, and L. Zhang, "Linear rate convergence of the alternating direction method of multipliers for convex composite programming," *Mathematics of Operations Research*, vol. 43, no. 2, pp. 622–637, 2018.
[58] D. Ruiz, "A scaling algorithm to equilibrate both rows and columns norms in matrices," CM-P00040415, Tech. Rep., 2001.
[59] T. Pock and A. Chambolle, "Diagonal preconditioning for first order primal-dual algorithms in convex optimization," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 1762–1769.
[60] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, Princeton, NJ, 1970.
[61] R. Wittmann, "Approximation of fixed points of nonexpansive mappings," *Archiv der mathematik*, vol. 58, pp. 486–491, 1992.

Qi Wang (Member, IEEE) received the B.S. degree in 2019 from the School of Electrical Engineering and Automation at Harbin Institute of Technology, and the Ph.D. degree in 2024 from the Department of Electrical Engineering at Tsinghua University. He is currently a Postdoc Fellow at the Hong Kong Polytechnic University. His research interests focus on efficient algorithms empowered by optimization and AI for the operation and control of energy systems. Dr. Wang serves as a Youth Editorial Board Member of Protection and Control of Modern Power Systems. He was awarded the Outstanding Doctoral Dissertation Award in 2024.

Guojun Zhang is a Ph.D. candidate in the Department of Applied Mathematics at The Hong Kong Polytechnic University. His research focuses on optimization theory and complexity analysis, computational optimal transport, and software development. His research has appeared in leading journals such as the SIAM Journal on Optimization, Mathematical Programming Computation, and IEEE Transactions on Pattern Analysis and Machine Intelligence.

Yue Yang (Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Tsinghua University in 2017 and 2022, respectively. He is currently an Assistant Professor with Hefei University of Technology, Anhui, China. His research interests include optimization and control in power system with integration of renewable energy.

Chao Ren (Member, IEEE) received the B.E. degree from the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China, in July 2017, and the Ph.D. degree from Interdisciplinary Graduate School, Nanyang Technological University, Singapore, in March 2022, holding Winner of Graduate College Research Excellence Award. Previously, he was a Wallenberg-NTU Presidential Postdoctoral Fellow at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Currently, he is a Wallenberg-NTU Presidential Researcher at the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Sweden. His research interests include machine learning, quantum computing, big data-analytics and their applications to smart grid. His work has been published more than 50 papers in top-tier journals and conferences, including TPWRS, JSAC, COMST, TDSC, TKDE, TSG, ICML, NeurIPS, AAAI, IJCAI, etc.

Wenchuan Wu (Fellow, IEEE) received B.S., M.S., and Ph.D. degrees from the Department of Electrical Engineering, Tsinghua University, Beijing, China. He is currently a Professor with Tsinghua University. His research interests include Energy Management Systems, active distribution system operations and control, machine learning and its application in energy systems. Prof. Wu was a recipient of the National Science Fund of China Distinguished Young Scholar Award in 2017.

Xinyuan Zhao is a Professor in the Department of Mathematics at Beijing University of Technology. She received her Ph.D. in Mathematics from the National University of Singapore in 2009, and her research focuses on large-scale optimization algorithms and high-performance solvers. She has led multiple research projects funded by the National Natural Science Foundation of China, Beijing municipal funding programs, and industry partners. In 2022, she received the Operations Research Application Award from the Operations Research Society of China. She serves as a Technical Editor for Mathematical Programming Computation (starting in 2025) and as an Associate Editor for the Asia Pacific Journal of Operational Research (since 2022), and she is the Deputy Secretary-General of both the Mathematical Programming Branch and the Mathematics and Intelligence Branch of the Operations Research Society of China.

Mikael Skoglund (Fellow, IEEE) received the Ph.D. degree in 1997 from Chalmers University of Technology, Sweden. In 1997, he joined the Royal Institute of Technology (KTH), Stockholm, Sweden, where he was appointed to the Chair in Communication Theory in 2003. At KTH he heads the Division of Information Science and Engineering.

Dr. Skoglund has worked on problems in source-channel coding, coding and transmission for wireless communications, Shannon theory, information-theoretic security, information theory for statistics and learning, information and control, and signal processing. He has authored and co-authored more than 200 journal and 450 conference papers in these fields.

Dr. Skoglund is a Fellow of the IEEE. During 2003–08 he was an associate editor for the IEEE Transactions on Communications. In the interval 2008–12 he was on the editorial board for the IEEE Transactions on Information Theory and starting in the Fall of 2021 he he joined it once again. He has served on numerous technical program committees for IEEE sponsored conferences, he was general co-chair for IEEE ITW 2019 and TPC co-chair for IEEE ISIT 2022. He is a member of the ELLIS Society, and an elected member of the IEEE Information Theory Society Board of Governors.

Defeng Sun is currently Chair Professor of Applied Optimization and Operations Research at the Hong Kong Polytechnic University. He was the President of the Hong Kong Mathematical Society. He mainly publishes in continuous optimization and machine learning. Together with Professor Kim-Chuan Toh and Dr Liuqin Yang, he was awarded the triennial 2018 Beale–Orchard-Hays Prize for Excellence in Computational Mathematical Programming by the Mathematical Optimization Society. In 2020, he was elected as a Fellow of the societies CSIAM and SIAM. In 2022, he received the RGC Senior Research Fellow Scheme award. He has also been named a Fellow of the AMS for 2026.