Electric Power Systems Research 212 (2022) 108613

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

BELTISTOS: A robust interior point method for large-scale optimal power flow problems☆

Juraj Kardoš [a], Drosos Kourounis [b], Olaf Schenk [a,*], Ray Zimmerman [c]

[a] Institute of Computing, Università della Svizzera italiana, Switzerland
[b] NEPLAN AG, Küsnacht Zürich, Switzerland
[c] Charles H. Dyson School of Applied Economics and Management, Cornell University, NY, USA

ARTICLE INFO

Keywords:
Interior point methods
Optimal power flow
Multi-period optimal power flow
Optimization software
Structure-exploiting algorithms
Performance profiling

ABSTRACT

Optimal power flow (OPF) problems are ubiquitous for daily power grid operations and planning. These optimal control problems are nonlinear, non-convex, and computationally demanding for large power networks especially for OPF problems defined over a large number of time periods, which are commonly intertemporally coupled due to constraints associated with energy storage devices. A robust interior point optimization library BELTISTOS is proposed, which allows fast and accurate solutions to single-period OPF problems and significantly accelerates the solution of multi-period OPF problems via the aid of structure-exploiting algorithms. Adhering to high reporting standards for replicable and reliable analysis, BELTISTOS is compared with interior point optimizers within the software package MATPOWER and evaluated using large scale power networks with up to 193,000 buses and problems spanning up to 4800 time periods.

1. Introduction

Since the formulation of alternating current (AC) optimal power flow (OPF) by Carpentier [1] as a continuous nonlinear programming (NLP) problem, OPF has become one of the most important and widely studied constrained nonlinear optimal control problems. It is concerned with optimization of an electric power network operation subject to physical constraints imposed by electrical laws and various engineering limits. Modern power grids increasingly adopt energy storage devices to address new challenges in the operation of the power grid associated with the adaptation of renewable energy sources [2,3]. Modeling of storage devices, however, introduces intertemporal couplings of the associated single-period OPF problems defined at each time period. The resulting multi-period OPF (MPOPF) problems [4], e.g., storage sizing and placement [5], become intractable for the general purpose NLP optimization methods due to the extensive memory and time requirements [6,7], thus limiting the effective use of optimization packages.

Interior point (IP) methods [8,9] are one of the solution methods for nonlinear and non-convex OPF problems. IP methods were applied to OPF problems by Quintana et al. [10], and recently studied by Capitanescu [11,12]. Various approaches and reformulations of the non-convex OPF problem have been proposed in order to reduce the computational complexity. Semi-definite programming has gained considerable interest in the literature of the last decade, where several extensions of these algorithmic approaches are reported in recent studies. The conditions necessary to recover a feasible solution of the AC OPF model from the optimal solution of the convex second-order cone are reported in [13]. An alternative to Ybus admittance matrix is proposed in [14], allowing for engineering-based convex relaxations. Finally, [15] proposes strengthening of AC OPF relaxations via analyzing cliques in the network graph. The resulting convex relaxation has a tight gap, yet the computing efficiency is much higher than the standard relaxation techniques. Modern trends in power grid operations and modeling, however, render approximation-based optimization techniques less attractive for coping with stressed operating conditions. Consequently, such relaxations do not aim at replacing non-convex solvers commonly converging to local minima, but rather complementing them providing estimates of appropriate metrics such as the optimality, the duality gap, and an infeasibility certificate at the converged solution.

Commonly proposed OPF solution algorithms are often presented and evaluated on a limited variety of power systems, neglecting the

☆ This research is part of the activities of the Innosuisse project no 34394.1 entitled "High-Performance Data Analytics Framework for Power Markets Simulation" and the Swiss Centre for Competence in Energy Research on the Future Swiss Electrical Infrastructure (SCCER-FURIES), which is financially supported by the Swiss Innovation Agency (Innosuisse - SCCER program).

* Corresponding author.
E-mail address: olaf.schenk@usi.ch (O. Schenk).

https://doi.org/10.1016/j.epsr.2022.108613
Received 12 November 2020; Received in revised form 23 February 2022; Accepted 3 July 2022
Available online 19 July 2022

effect of NLP solvers or sparse linear algebra kernels adopted in the optimization method; see, e.g., [16,17]. A study presented in [17] performs benchmarks on three different OPF formulations (polar-power, rectangular-power, and rectangular-current) and seven test power networks (ranging from 118 to 3120 buses), considering also various initializations. The performance is reported using performance profiles for five NLP solvers, IPOPT, KNITRO and other active set methods with Hessian approximations, which are not well-suited for the large-scale nonlinear OPF problems. Default settings for the optimizers are selected, acknowledging the fact that customizing the optimizers options instead of using the default parameters could result in a very different outcome. The authors conclude that IPOPT is faster overall than the other NLP solvers. However, this study neglects the significance of sparse linear algebra kernels that are of extreme importance for the IP methods, especially for large-scale realistic power networks where the associated linear systems may be significantly more ill-conditioned due to different scales of modeling and equipment characteristic values.

IP methods demonstrate polynomial time asymptotic complexity and speed of convergence, which often exhibits superlinear and quadratic asymptotic rates [18]. Another advantage of IP methods is that they allow for a variety of different direct sparse or iterative solution strategies for the underlying linear systems obtained from the linearization of the optimality conditions at each iteration. The linear system solution is the most computationally expensive task of an interior point iteration, especially for large power networks or MPOPF problems. Additionally, robust sparse linear algebra kernels are of extreme importance for obtaining accurate search directions, especially for realistic power networks, where the associated linear systems may be highly ill-conditioned due to different scales of model parameters and equipment characteristic values. The overall efficacy of the IP algorithm thus relies on fast and reliable linear system solvers.

In this work, a robust solution strategy for the linear systems stemming from IP algorithms is introduced. The method is based on multi-level inverse-based factorization combined with Krylov subspace method. The reliability of the proposed method, implemented within the BELTISTOS library [19], is demonstrated for a variety of benchmark scenarios including (i) various OPF formulations, (ii) different initializations of the IP method, and (iii) comparison with open source and commercial NLP solvers. Furthermore, (iv) additional emphasis is given on the influence of sparse linear algebra components to the overall convergence and performance of the IP method. Finally, (v) all these aspects are studied on networks of increasing complexity and sizes. Another integral part of BELTISTOS are structure-exploiting algorithms applicable to coupled MPOPF problems. BELTISTOS is integrated into general purpose power system software, such as MATLAB-based MATPOWER [20,21], or it can be easily used in Julia-based PowerModels.jl [16]. This provides researchers and educators a platform for solving an extensible collection of OPF problem formulations directly applicable to power networks of increasing complexity.

The rest of this paper is organized as follows. Section 2 summarizes the mathematical formulation of OPF and MPOPF problems. In Section 3, IP methods are the underlying computational strategies for the linear systems are introduced. Section 4 presents the experimental setup, while numerical results for the OPF problems are summarized in Section 5. The results for the MPOPF benchmarks are summarized in Section 6, while the final remarks are presented in Section 7.

2. OPF problems

The AC OPF [20,21], formulated as a NLP problem (1), aims to find the optimal settings of generator powers and bus voltages in order to minimize a nonlinear function $f(x)$ representing the power generation costs. The feasible region is defined by equality constraints $c_E(x)$ representing nodal power balance equations, inequality constraints $c_I(x)$ representing the transmission line power flow limits, and rectangular bounds on $x_{\min}$, $x_{\max}$ on the control variables including voltage magnitudes V_m, and active and reactive generator injections P_g and Q_g. The NLP problem reads

$$\underset{x}{\text{minimize}} \quad f(x) \tag{1a}$$

$$\text{subject to} \quad c_E(x) = 0, \tag{1b}$$

$$c_I(x) \le 0, \tag{1c}$$

$$x_{\min} \le x \le x_{\max}. \tag{1d}$$

In the following discussion, n_b, n_g and n_l denote the number of buses, generators and transmission lines, respectively. The optimization variables $x \in \mathbb{R}^{N_x}$ for the standard AC OPF problem consists of the $n_b \times 1$ vectors of voltage angles Θ and magnitudes V_m and the $n_g \times 1$ vectors of generator real and reactive power injections P_g and Q_g. The complex voltage in polar coordinates is defined as $V = V_m e^{j\Theta}$. Equality constraints, (1b), involve complex power balance equations which are split into a set of $2n_b$ nonlinear and nonconvex equations for its real and reactive parts. The inequality constraints (1c) consist of two sets of n_l branch flow limits expressed as nonlinear functions of the bus voltage angles and magnitudes, one for the each end of the branch.

The AC OPF problem takes different forms based on the different representations of the complex bus voltages V, which can be represented either in rectangular or polar coordinates. The optimization vector x, considering the rectangular coordinates $V = U + jW$, takes the form $x = \begin{bmatrix} U & W & P_g & Q_g \end{bmatrix}$. Another variation of the standard AC OPF problem uses current balance constraints in place of the power balance constraints (1b). Further details, comprehensive explanation of the variables, nodal balance equations and discussion regarding the modeling aspects, are described in the MATPOWER OPF model [20,21].

2.1. Multi-period OPF problems

Multi-period OPF problems couple the individual single-period AC OPF problems (1) over multiple time periods $n = 1, 2, \ldots, N$. The OPF constraints (1b)–(1d) must hold at each time period n and are independent of each other. The inter-temporal coupling is introduced by the equations modeling the storage energy levels over the planning time horizon. The variables P_g are extended by the power output of the storage devices $P_g^S = \begin{bmatrix} P_g^{Sd} & P_g^{Sc} \end{bmatrix} \in \mathbb{R}^{2N_S N}$ including discharging and charging powers, respectively. The evolution of the vector of storage energy levels $\epsilon_n \in \mathbb{R}^{N_S}$, where N_S represents the number of storage devices, follows the update equation

$$\epsilon_n = \epsilon_{n-1} + B^S \left(P_g^S\right)_n, \quad n = 1, \ldots, N. \tag{2}$$

The energy level at each period needs to honor minimum and maximum storage capacity levels $\epsilon_{\min}$ and $\epsilon_{\max}$. The initial storage level is denoted ϵ_0 and the constant matrix $B^S \in \mathbb{R}^{N_S \times 2N_S}$ models the discharging and charging efficiencies η_d and η_c of the storage device. The resulting linear inequality constraints extend the constraint set of the coupled single-period OPF problems, reading

$$\underset{x_1,\ldots,x_N}{\text{minimize}} \quad \sum_{n=1}^{N} f(x_n) \tag{3a}$$

$$\text{subject to} \quad \forall n = 1, 2, \ldots, N: \quad c_E(x_n) = 0, \tag{3b}$$

$$c_I(x_n) \le 0, \tag{3c}$$

$$\epsilon_{\min} \le \epsilon_n \le \epsilon_{\max}, \tag{3d}$$

$$x_{\min} \le x_n \le x_{\max}. \tag{3e}$$

Comprehensive MPOPF model description used in this work, together with the model source code, is available online [19,22].

Standard solutions strategies for the MPOPF problems adopt general purpose NLP methods. However, the problem size grows linearly with increasing length of the time horizon while the computational complexity for factorizing and solving the underlying linear systems at each IP iteration is cubic and quadratic, respectively. The overall problem thus becomes computationally intractable for large power networks spanning long time horizons. A wide-spread solution heuristic to solve the MPOPF problems is to adopt a decomposition strategy based on splitting the long time horizon into smaller intervals using receding horizon control strategies [23]. Such solutions, however, become suboptimal compared to the fully coupled solution. An algorithm for solving the fully coupled AC MPOPF problems was first presented in the previous work [4,24].

3. Interior point methods

Primal–dual IP methods [25,26] approach the NLP problem (1) by first transforming the inequality constraints to the equalities using additional slack variables $s \in \mathbb{R}^{N_s}$ in addition to the unknowns $x \in \mathbb{R}^{N_x}$. Subsequently, they solve a sequence of barrier μ-subproblems, which are obtained by adding logarithmic barrier terms to the objective function to ensure their non-negativity. Each μ-subproblem is solved only approximately, decreasing the barrier parameter μ during the iterations, reaching the value close to zero as the optimal point of the original problem is approached. The solution of each μ-subproblem is a critical point of the Lagrangian $L(x, s, \lambda_E, \lambda_I)$ where λ_E, λ_I are the vectors representing the Lagrange multipliers for the equality and inequality constraints. The Lagrangian is formulated as

$$L := f(x) - \mu \sum_{i=1}^{N_x} \ln(x_i) - \mu \sum_{i=1}^{N_s} \ln(s_i) + \lambda_E^\intercal c_E(x) + \lambda_I^\intercal (c_I(x) - s). \tag{4}$$

The solutions satisfy the perturbed Karush–Kuhn–Tucker (KKT) optimality conditions. Primal–dual interior point methods define the dual variables $y \equiv \frac{\mu}{s}$ and $z \equiv \frac{\mu}{x}$ augmenting the optimality conditions

$$\begin{bmatrix} \nabla_x L \\ \nabla_s L \\ \nabla_{\lambda_E} L \\ \nabla_{\lambda_I} L \\ \nabla_z L \\ \nabla_y L \end{bmatrix} = \begin{bmatrix} \nabla_x f(x) + J_E^\intercal \lambda_E + J_I^\intercal \lambda_I - z \\ y + \lambda_I \\ c_E(x) \\ c_I(x) - s \\ Zx - \mu e \\ Ys - \mu e \end{bmatrix}, \tag{5}$$

where $X = \mathrm{diag}(x)$, $S = \mathrm{diag}(s)$, $Y = \mathrm{diag}(y)$, $Z = \mathrm{diag}(z)$ and e is an unit vector of appropriate size. $J_E = \nabla_x c_E(x)$ and $J_I = \nabla_x c_I(x)$ are Jacobians of the equality and inequality constraints, respectively. The primal–dual update is obtained from the linearization of the perturbed KKT conditions. Commonly, equations related to the perturbed dual variables are first eliminated to form a reduced symmetric linear system $K\Delta u = b$ solved at each IP iteration, where $K \in \mathbb{R}^{n_k, n_k}$ is a sparse indefinite matrix, Δu is the unknown Newton direction, and b is the associated right hand side,

$$K = \begin{bmatrix} \tilde{H} & 0 & J_E^\intercal & J_I^\intercal \\ 0 & L_s & 0 & -I \\ J_E & 0 & 0 & 0 \\ J_I & -I & 0 & 0 \end{bmatrix}, \quad \Delta u = \begin{bmatrix} \Delta x \\ \Delta s \\ \Delta \lambda_E \\ \Delta \lambda_I \end{bmatrix}, \tag{6}$$

where $\tilde{H} = \nabla^2_{xx} L + X^{-1} Z$, $L_s = S^{-1} Y$. For additional details on the IP algorithm, the interested reader is referred to [25,26] and the classical textbook [27]. The performance of the IP method for large scale problems relies on accurate, fast and memory efficient sparse linear solvers, since the resulting linear systems are commonly sparse, large and ill-conditioned [6].

3.1. Single-period OPF solution strategy

A widespread approach for solving KKT systems of form (6) consists of employing general purpose direct sparse multifrontal methods. The direct sparse solvers, such as PARDISO [28] or MA57 [29], obtain the solution of the linear system by LDL^T factorization and subsequent forward–backward substitutions [30]. However, a straightforward application of these techniques to highly ill-conditioned KKT systems can result in insufficient accuracy of the solution or infeasible computational times due to more expensive pivoting schemes [31].

An alternative approach adopts a modern and robust sparse elimination methodology for large-scale OPF problems based on inverse-based pivoting strategy [31–33]. A key component of this inverse-based pivoting approach is driven by a preprocessing phase based on combinatorial algorithms which improve diagonal dominance, reduce fill-in and increase concurrency to allow for parallel treatment. Moreover, these inverse-based pivoting methods also allow the detection of dense submatrices, which can utilize optimized dense linear algebra kernels provided by the Intel Math Kernel library (MKL) [34]. The permutation strategy is based on maximum weighted matchings in a bipartite graph associated with the KKT matrix. The matching associated with the permutation is sought, such that the diagonal dominance of the permuted matrix is maximized. The second objective for the matching is to preserve the symmetry of the matrix, which is achieved by following the cycles in the permutation matrix associated with the non-symmetric maximum weighted matching. After the system $K \in \mathbb{R}^{n_k, n_k}$ is reordered $P_S^T D K D P_S = \bar{K}$, where $D, P_S \in \mathbb{R}^{n_k, n_k}$ are diagonal and permutation matrices, respectively, the system $\bar{K}$ will have many well-conditioned diagonal blocks of size 1×1 and 2×2. Subsequently, the associated block graph of $\bar{K}$ is reordered by a symmetric reordering, such that $\Pi^T \bar{K} \Pi = \tilde{K}$, with the symmetric block permutation $\Pi \in \mathbb{R}^{n_k, n_k}$, and the block structure

$$\tilde{K} = \begin{bmatrix} G & F^T \\ F & C \end{bmatrix}, \tag{7}$$

where the blocks are $G \in \mathbb{R}^{k,k}, F \in \mathbb{R}^{n_k - k, k}$ and $C \in \mathbb{R}^{n_k - k, n_k - k}$ such that $k \leq n_k$. During the factorization, the pivots might still be too small, therefore such update is postponed and the corresponding rows and columns are symmetrically permuted to the end, forming a Schur complement block. The inverse based pivoting strategy computes an incomplete factorization $\tilde{K} \approx LDL^T$, such that

$$\tilde{K} \approx \underbrace{\begin{bmatrix} L_G & 0 \\ L_F & I \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} D_G & 0 \\ 0 & S_C \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} L_G^T & L_F^T \\ 0 & I \end{bmatrix}}_{L^T}, \tag{8}$$

where $L_G, D_G \in \mathbb{R}^{k,k}$ are lower triangular with unit diagonal and block diagonal with blocks of sizes 1×1 and 2×2, respectively. L_F is a factor of size $\mathbb{R}^{n_k - k, k}$, $S_C \in \mathbb{R}^{n_k - k, n_n - k}$ is the Schur complement $S_C = C - L_F D_G L_F^T$, and I is an identity matrix. The delayed pivots in the Schur complement block will be computed explicitly using direct factorization techniques, such as the method implemented in PARDISO or the multilevel inverse based strategy can be applied recursively. The Schur complement blocks are visualized in Fig. 1 illustrating the KKT matrix and the three-level $LDL^\intercal$ factors for the IEEE 300 bus test system. The key advantage of these methods is that small pivots will be detected during the elimination phase. As a result the linear systems are reordered so that the largest entries in magnitude appear near the main diagonal of the matrix and the pivots are postponed whenever the norm of the factor $\|L^{-1}\|$ exceeds a prescribed bound.

Finally, in order to further improve accuracy and robustness, a strategy based on symmetric quasi-minimal residual Krylov subspace method [35,36] is used where the full multilevel incomplete LDL^T factors are used as a preconditioner. Further algorithmic details are provided in the Refs. [31,32,37].

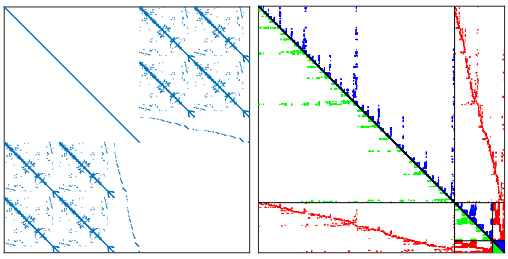


Fig. 1. Matrix structure. Left: KKT matrix for IEEE 300 bus system, Right: Three-level inverse based matrix factor.

3.2. Structure exploiting algorithm for MPOPF

The factorization or large KKT systems is a computationally expensive operation introducing significant fill-in, which may quickly exhaust the memory installed on shared memory machines for large-scale MPOPF problems. Furthermore, the general-purpose factorization codes are not aware of the underlying structural properties of the particular KKT systems associated with MPOPF problems [4,38] The appropriate structure of the KKT system for the MPOPF problem emerges from the fact that each of the variables in the optimization vector $\Delta u = \left[\Delta x, \Delta s, \Delta\lambda_E, \Delta\lambda_I\right]$ corresponds to some time period $n = 1, 2, \ldots, N$ of the MPOPF problem. The primal variables can be partitioned as $\Delta x = \left[\Delta x_1, \Delta x_2, \ldots, \Delta x_N\right]$. For example, consider the polar-based OPF formulations. The vector of primal variables for each time period Δx_n consists of the bus voltage angles and magnitudes and active and reactive power injections from the conventional generator and storage devices, i.e. $\Delta x_n = \Delta\left[\Theta, V_m, P_g, P_g^S, Q_g, Q_g^S\right]_n$, describing the power grid state in the nth time period. The equality constraints and the corresponding Lagrange multipliers can be partitioned in a similar fashion $\Delta\lambda_E = \left[\Delta\lambda_{E1}, \Delta\lambda_{E2}, \ldots, \Delta\lambda_{EN}\right]$. Inequality constraints $\Delta\lambda_I$ and the associated slack variables Δs follow a similar ordering. In order to reveal the scenario-local structure of the KKT system (6), the variables corresponding to the same time period n are grouped together, i.e., $\Delta\hat{u}_n = \left[\Delta x_n, \Delta\lambda_{En}, \Delta\lambda_{In}, \Delta s_n\right]$. Thus, the global ordering will be $\Delta\hat{u} = \left[\Delta\hat{u}_1, \ldots, \Delta\hat{u}_N, \Delta\hat{u}_c\right]$, where the coupling variables $\Delta\hat{u}_c = \Delta\lambda_A \subset \Delta\lambda_I$, representing the Lagrange multipliers corresponding to the linear constraints (3d), are placed at the end of the new optimization vector $\Delta\hat{u}$. The permutation matrix P can be introduced, such that $\Delta\hat{u} = P\Delta u$ Under the new permutation P, the KKT matrix from (6) obtains an arrowhead structure

$$\underbrace{\begin{bmatrix} A_1 & & & B_1^{\mathsf{T}} \\ & \ddots & & \vdots \\ & & A_N & B_N^{\mathsf{T}} \\ B_1 & \cdots & B_N & 0 \end{bmatrix}}_{\hat{K}=P^{\mathsf{T}}KP} \underbrace{\begin{bmatrix} \Delta\hat{u}_1 \\ \vdots \\ \Delta\hat{u}_N \\ \Delta\hat{u}_c \end{bmatrix}}_{\Delta\hat{u}} = \underbrace{\begin{bmatrix} b_1 \\ \vdots \\ b_N \\ b_c \end{bmatrix}}_{\hat{b}}. \tag{9}$$

The arrowhead system (9) can be solved more efficiently using decomposition algorithms based on Schur complement [4,6], shown in Algorithm 1, compared to the direct sparse solvers applied to the KKT system (6) in black-box fashion. The BELTISTOS algorithm solves the arrowhead system by factorizing the diagonal blocks A_n (line 5) and subsequently uses the factors to evaluate the contribution S_n to the Schur complement S_c of time period n (line 6). The factors are also used during the solve for the vector y_n (line 8) that contributes to the RHS r_C (line 9) of the dense solve (line 12). Finally, the vector $\Delta\hat{u}_C$ is used to form the RHS r_n (line 14) for the final solve (line 15) with the diagonal blocks A_n and their factors to obtain the desired solution vector $\Delta\hat{u}$.

Additionally, the solution of the dense Schur complement systems S_c arising in the MPOPF problems can be further accelerated by exploiting their particular structure [4].

Algorithm 1: BELTISTOS ALGORITHM.

```
1:  function SchurSolve(K̂, b̂)
2:      S_c := 0
3:      r_c := b_c
4:      for n = 1 : N do
5:          [L_n, D_n] := SparseFactorize(A_n)
6:          S_n := −B_n A_n^{−1} B_n^T
7:          S_c := S_c + S_n
8:          y_n := SparseSolve(L_n, D_n, b_n)
9:          r_c := r_c − B_n y_n
10:     end for
11:     [L_c, D_c] := DenseFactorize(S_c)
12:     Δû_c := DenseSolve(L_c, D_c, r_c)
13:     for n = 1 : N do
14:         r_n := b_n − B_n^T Δu_c
15:         Δû_n := SparseSolve(L_n, D_n, r_n)
16:     end for
17:     return Δû
18: end function
```

In case when the memory is bottleneck, e.g. when N is large, the memory requirements can become prohibitive. Algorithm 1 stores sparse factorizations L_n, D_n of all N diagonal blocks A_n in the memory. However, the memory requirements can be reduced by storing the factorization of the diagonal blocks A_n at line 5 only temporarily, keeping one set of the factorizations at a time in the memory. This means that the factorization of A_n blocks needs to be computed twice. First, as shown at line 5 and the second time before the evaluation of $\Delta\hat{u}_n$ in the second *for* loop at line 15.

3.3. Optimization software

In what follows, several different primal–dual IP optimization software tools are introduced. These are used by power grid practitioners for OPF problems as they are supported by the modeling tools such as MATPOWER of PowerModels.jl. The optimization software is summarized in Table 1, where the commercial solvers are highlighted. KNITRO [39] implements the IP algorithm with direct step computation designed for solving large-scale mathematical NLP problems. However, KNITRO may automatically switch to the iterative conjugate gradient (CG) algorithm if the direct step is suspected to be of poor quality. IPOPT [25,26] implements a primal–dual IP method for large-scale nonlinear optimization. Benchmarks have shown that MA57 [29] and MKL-PARDISO [28] are often the most reliable direct sparse linear solvers within IPOPT. The former is used in the single period OPF study, while the latter is used for the MPOPF problems. MIPS [40,41] is a primal–dual interior point solver for OPF problems. It is entirely implemented in MATLAB code and distributed with MATPOWER. FMINCON [42] is a part of the MATLAB optimization toolbox, providing an IP method and applies the projected conjugate gradient (PCG) method to solve the KKT system in an iterative fashion. BELTISTOS [19] is a suite of high-performance NLP solution methods for OPF algorithms [4,7,24] including extremely scalable and low memory MPOPF. BELTISTOS adopts selected algorithms implemented in IPOPT, adjusted specifically for the nature of the OPF problems, including techniques introduced in Section 3.1. BELTISTOS also implements structure exploiting and data compression algorithms designed for the particular structure of the MPOPF problems introduced in Section 3.2.

4. Benchmarks setup

Various characteristics of benchmark cases used in the following numerical experiments are listed in an accompanying technical report [22]. These are the cases distributed with MATPOWER, including

Table 1
Open source and commercial (highlighted) optimizers.

Optimizer	Version	Solver	License	Reference
BELTISTOS	1.0	PARDISO	Free academic use	[4,7,19,24]
KNITRO	12.2.0	MA57, CG	Artelys	[39]
IPOPT	3.12.10	MA57, MKL-PARDISO	Open source (EPL)	[25,26]
MIPS	1.3.1	'\' (MA57)	Open source (BSD)	[40,41]
FMINCON	2018b	PCG	MATLAB	[42]

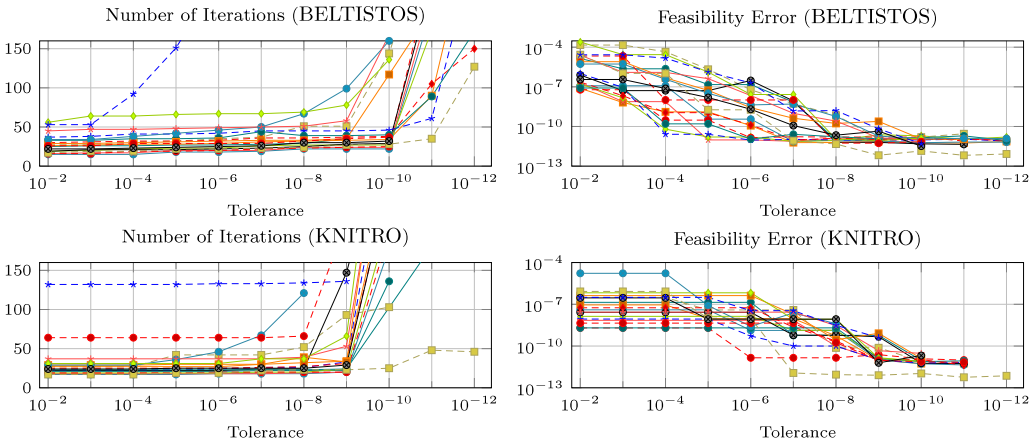


Fig. 2. Statistics for the standard benchmark cases (each line corresponds to one particular case listed in [22]).

snapshots of the full French very high-voltage and high-voltage grid (case labels suffixed with "rte") and pan-European fictitious data set (labeled as "PEGASE") [43,44]. A set of entirely synthetic cases (labeled as "ACTIVS") is geographically situated in US Northeast and Mid-Atlantic regions. The case is designed with a transmission network to serve a load that roughly mimics the actual population of its geographic footprint. The synthetic transmission system was designed by algorithms described in [45] to be statistically similar to actual transmission system models. The Polish networks during the winter/summer peak/off-peak conditions are represented by cases such as 2746wop and case3120sp [21] (the suffix representing the season and peak/off-peak conditions). In addition, there are four large-scale cases, case21k–case193k, built from the Polish system winter 2007–08 evening peak power flow data (case3012wp), considering the largest generator outage and line contingencies.[1] The benchmark cases were selected based on the system size and the number of variables involved in the OPF problem. Specifically, the OPF problem has to consist of at least 5,000 primal variables. The benchmark software suite along with all data sets and detailed instructions can be found in [22] and online [19]. Simulations are performed on a workstation equipped with an Intel Xeon CPU E7-4880 v2 at 2.50 GHz and 1 TB RAM using the MATPOWER version 7.0 [21]. The common factors influencing the performance were kept fixed. The fixed factors include: (i) the convergence tolerance of the optimization solvers, (ii) the initial point, and (iii) explicit sequential (single-core) execution. For all the NLP solvers the optimality, feasibility and complementarity tolerances were fixed at 10^{-6} and no more than 500 iterations where allowed while also enforcing a maximum time limit of 5 h.

4.1. Convergence tolerance

Before proceeding with the benchmarks, the selection of the convergence tolerance is analyzed. First of all, the convergence tests implemented by optimizers vary in some aspects, e.g. scaling of the residual errors or type of the norms, making the user specified tolerance not necessarily equivalent amongst the optimizers. Second, various stopping tolerances were used in previous OPF studies, ranging from 10^{-3} to 10^{-8} [46]. The selection of the convergence criteria has to consider multiple factors, including required optimality and feasibility errors, as well as account for numerical issues associated with tight tolerances due to the ill-conditioning leading to inaccurate search directions and thus stagnating convergence. For the experiments in this section, the "acceptable" termination criteria was disabled. This does not allow the algorithm to terminate before the desired convergence tolerance is met (e.g. in case there is no improvement in the objective function or feasibility over some specified number of iterations). It can be observed that there is no significant improvement of the objective function value for the tight tolerances. The absolute and relative errors of the objective function value between tolerances 10^{-4} and 10^{-9} are less than order of 10^{-3} and 10^{-9}, respectively. In several cases, the objective function slightly increased for tighter tolerances, in similar orders of magnitude. The constraint violations (feasibility error) and number of iterations for different tolerances using BELTISTOS and KNITRO are shown in Fig. 2. The figure illustrates that even for a modest tolerance of 10^{-4} the constraint violations are usually much smaller and in most cases tightening the tolerance further will result in only a few additional iterations. For tight tolerances below 10^{-9}, convergence stagnates and the number of iterations starts to significantly grow (e.g. KNITRO) or optimizers terminate with an error message. Some optimizers were not able to reach the tight tolerances, e.g. IPOPT and MIPS failed to solve most of the benchmarks for tolerances below 10^{-9}, while FMINCON terminated with feasibility errors larger than the specified tolerance. A tolerance value 10^{-6} is thus used in order to focus on the performance of the optimizer, and isolate issues related to the numerical errors.

4.2. Initial point

Selection of a high-quality initial point is crucial for gradient-based optimization methods, especially when applied to nonconvex AC OPF problems. In this study, the solution of the power flow equations is used for initialization, as opposed to other heuristically chosen mid points (average of upper and lower bounds). Up to 10-fold performance improvement was observed, while the solution success rate improved by up to 20% for such initial points. An extended discussion is available in the technical report [22].

[1] Data available at http://www.beltistos.com/.

Table 2
Number of solved benchmarks out of twenty-five test cases for different OPF formulations.

Optimizer	Polar Power	Rect. Power	Polar Curr.	Rect. Curr.	Total
BELTISTOS	25	25	25	25	100
KNITRO	25	25	25	24	99
FMINCON	20	24	20	24	88
IPOPT	21	21	23	21	86
MIPS	22	19	21	17	79

Table 3
Overall time (s) for large-scale benchmarks (dash represents failure or time limit). Polar-Power formulation.

Benchmark	BELTISTOS	KNITRO	MIPS	IPOPT	FMINCON
ACTIVSg25k	43.04	45.79	108.18	50.06	183.57
ACTIVSg70k	152.38	187.65	—	152.56	644.30
case21k	65.27	47.05	241.20	—	1,548.28
case42k	220.83	161.59	1,957.95	—	—
case99k	1,231.90	753.07	—	—	17,325.98
case193k	4,360.57	1,889.51	—	—	—

4.3. Performance profiles

Performance profiles [47,48] are used for a compact comparison and evaluation of different methods such as optimization methods or OPF problems formulations on a set of benchmark problems. The profiles are generated by running the set of methods $\mathcal{M}$ on a set of problems $\mathcal{P}$ and recording information of interest, e.g., time to solution or memory consumption. Consider that the method $m \in \mathcal{M}$ reports a statistic $\theta_{mp} \geq 0$ for the problem $p \in \mathcal{P}$, where the smaller statistics θ_{mp} indicate better solution strategies. The best statistic for a given problem p is defined as $\theta_p^* = \min_{m\in\mathcal{M}}\{\theta_{mp}\}$. Then for $\alpha \geq 1$ and each $m \in \mathcal{M}$ and $p \in \mathcal{P}$ the indicator k is defined

$$k(\theta_{mp}, \theta_p^*, \alpha) = \begin{cases} 1 & \theta_{mp} \leq \alpha \cdot \theta_p^*, \\ 0 & \theta_{mp} > \alpha \cdot \theta_p^*. \end{cases} \tag{10}$$

The performance profile $p_m(\alpha)$ of the method m is then defined by

$$p_m(\alpha) = \frac{\sum_{p\in\mathcal{P}} k(\theta_{mp}, \theta_p^*, \alpha)}{|\mathcal{P}|}. \tag{11}$$

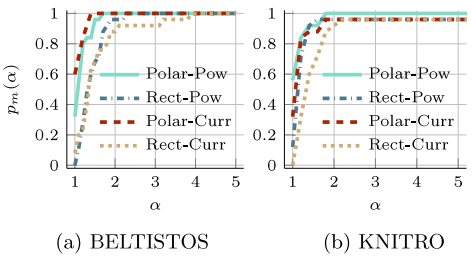
Thus, the value of $p_m(\alpha)$ indicates the fraction of all examples which can be solved within a factor of α of the best solver time. The fraction of problems on which optimizer m is the most effective is given by $p_m(1)$ and $p_m^* := \lim_{\alpha\to\infty} p_m(\alpha)$ indicates the fraction for which the optimizer successfully found the solution.

5. Numerical results - OPF

5.1. OPF formulations

As introduced in Section 2, different representations of the voltage variables or the nodal equations can be used to formulate the OPF problem, which results in constraints with different nonlinearities and feasible regions. Thus, the constraint Jacobians as well as their sparsity structures and condition numbers will vary, accelerating or delaying the convergence.

Table 2 provides a summary of the optimizer success rates for solving different OPF formulations. The table shows that robust optimizers such as BELTISTOS or KNITRO are marginally influenced by the choice of the formulation. Similar holds for IPOPT, but it fails to solve most of the large-scale cases. FMINCON solves more cases with rectangular-based voltage formulations, while MIPS provides better success rate for the polar voltages. The optimizers converged to the same solution for all OPF formulations, up to the maximum relative error 10^{-5}. The performance profiles for the OPF formulations for various optimizers are shown in Fig. 3.



(a) BELTISTOS (b) KNITRO

Fig. 3. Overall time profiles for various OPF formulations.

BELTISTOS, solves the polar-based voltage formulations slightly easier than other formulations, resulting in up to four times improvements in terms of the overall time, with similar improvement in the iteration count. The memory requirements are slightly increased for Rectangular formulations, requiring up to 20% more memory (9.4 GB for the largest benchmark). Similar results are observed for KNITRO. In average, the overall time of Rectangular-Current formulation was increased up to 30% compared to the other formulations, which performed very similarly. The memory usage of KNITRO was increased compared to BELTISTOS, requiring up to 10 GB. FMINCON converges faster, in general, using power-based formulations.

5.2. Performance evaluation

Robustness, performance, and memory efficiency of all optimization software packages is the main focus in this section.

Table 3 shows the summary of overall solution times for the large-scale test cases and all optimizers using the default Polar-Power formulation (an extensive list of all results can be found in the technical report [22]).

The performance profiles for all optimization packages considering all benchmarks cases are shown in Figs. 4(a)–4(d). The performance profiles clearly indicate that BELTISTOS and KNITRO converged to the optimal solution for the majority of the benchmark cases, with slight advantage of BELTISTOS in terms of robustness. BELTISTOS was faster for 34% of the benchmark cases, KNITRO being slower by up to a factor or 50.9 and 5.3 in the two extreme cases. BELTISTOS was no more than 50% slower in 86% of the benchmark cases. In the two worst cases, BELTISTOS was slower by a factor of 3.2 and 2.3, respectively. IPOPT optimizer was not competitive for the solution of the large-scale cases, both in terms of robustness and performance, failing for the 58% of the large-scale cases, not being able to solve the four largest cases as can be seen in Table 3. The performance and success rate of MIPS and FMINCON behaves differently depending on the OPF formulation. Considering the FMINCON's top performing Rectangular-Power OPF formulation, it is still in average slower by a factor of 4–5 compared to KNITRO and BELTISTOS, with the maximum slowdown up to a factor of 12.5.

In terms of memory efficiency, the best results are achieved by MIPS and KNITRO, closely followed by BELTISTOS. The difference in the memory usage between these three optimization packages varies by up to a factor of 2.2.

5.3. BELTISTOS OPF solution strategy

BELTISTOS utilizes the robust multilevel inverse-based factorization solver introduced in Section 3.1. The accuracy of the multilevel solution strategy for the KKT system at each iteration of the IP method is illustrated in Fig. 5, comparing it to a standard direct sparse solver MKL-PARDISO used in IPOPT. Three benchmark cases of increasing size are illustrated, providing insight into the numerical error for the solution of the KKT system in each IP iteration.

The multilevel strategy implemented in BELTISTOS consistently solves the KKT systems with numerical error close to the machine

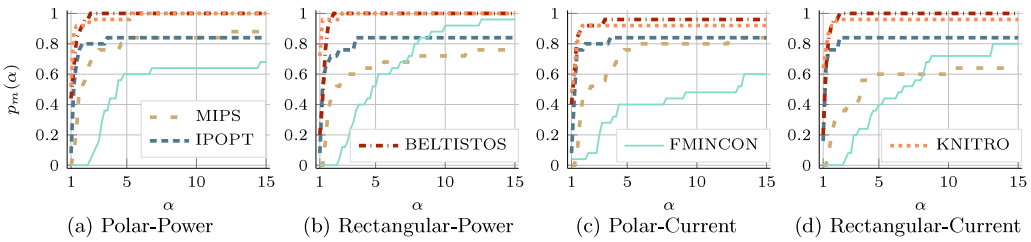


Fig. 4. Overall time performance profiles for OPF formulations considering all benchmarks.

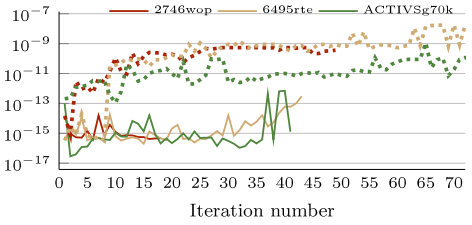


Fig. 5. Residuals for BELTISTOS multilevel strategy (solid) and IPOPT (dotted).

precision, which is an improvement by up to 5–6 orders of magnitude compared to IPOPT. Note that BELTISTOS converged within 20–40 iterations for the three benchmarks, while IPOPT required up to 70 iterations to converge to the desired tolerance. It is evident that more accurate search directions, obtained with the multilevel strategy, reduce the overall number of iterations needed until convergence, and thus the overall solution time of the IP algorithm, as well as improve its robustness and numerical stability.

BELTISTOS and KNITRO demonstrate superior performance, both in terms of solution time and iteration counts, compared to the other optimizers. Additionally, KNITRO performs better than BELTISTOS for 4 out of 6 large-scale testbeds in the polar-power formulation for the single period benchmarks. However, the clear advantage of BELTISTOS for solving the multi-period problems is detailed in the next section.

6. Numerical results - MPOPF

The computational complexity of the MPOPF problems grows quickly with the increasing number of time periods. In this section, the value of a structure exploiting algorithm is demonstrated compared to a general purpose optimization methods such as IPOPT (with MKL-PARDISO linear solver) or KNITRO. The benchmarks in this section focus on performance related to the factorization and backsolve phases, since these represent the bottleneck of the IP method for large-scale MPOPF problems. Common factors for both IP methods are not included in the measurements, including computations such as evaluation of the Jacobian or Hessian matrices. The benchmark cases include one small (IEEE 118) and two medium sized (PEGASE 1,354 and 2,869 [44]) power networks, considering up to $N_s = 100$ storage devices and N =4800 time periods. The largest problem size contains 32 million variables and 72 million nonlinear constraints.

6.1. Number of time periods and storage devices

Performance of the optimizers is investigated for increasing problem sizes by changing the number of time periods and storage devices. The average time of the KKT system solution is shown in Fig. 6. It is evident that BELTISTOS outperforms the general purpose solution approach, providing orders of magnitude faster solution times. A comparison of the factorization and forward–backward substitution phases is also shown, illustrated by the horizontal line inside the bars (note the logarithmic scale on the y-axis). The factorization phase clearly dominates for IPOPT and KNITRO, therefore the forward–backward substitution phase is not visible in the figure. The factorization and the backsubstitution phases are comparable for BELTISTOS, with the backsubstitution phase taking more time for the memory efficient version, since some portion of the computation needs to be recomputed in order to keep the memory requirements as low as possible. However, the overall performance benefit is still significant, compared to structure unaware solution methods. The performance gap between BELTISTOS and other optimizers increases with increasing values of N. For $N = 4800$ IPOPT failed due to exceeding the memory limit during the symbolic factorization of matrix for the smallest benchmark, with similar behavior also for KNITRO. With the increasing benchmark size, the failure was observed also for $N = 2400$ or $N = 1200$ due to exceeding the available memory or the time limit. BELTISTOS requires approximately 1% of the time needed by the best competitor for the smallest problem, with increasing performance benefit for larger problems.

Furthermore, the performance of BELTISTOS is analyzed for increasing number of storage devices. Such problems arise in storage sizing and placement problems, where the optimal sizes and locations of the storage devices are sought. The number of storage devices increases the number of coupling variables and thus the size of the Schur complement, therefore also posing a bottleneck for large problems. Fig. 7 shows the average solution time of the KKT system for increasing number of storage devices N_s for different power grids, considering $N = 600$ time periods. In case of the PEGASE 1,354 bus benchmark, ten-fold increase in the storage device number resulted in 81 times longer computation for IPOPT, compared to a 3.7 times increase for the computation time of BELTISTOS.

6.2. Memory resources

Fig. 8 illustrates the memory requirements of each solver on the same set of MPOPF benchmarks as in Fig. 6. Clearly, the general purpose approach relying on a direct sparse solver requires prohibitive memory resources due to excessive fill in. The structure-exploiting approach implemented in BELTISTOS reduces the memory requirements by more than one order of magnitude by usage of efficient linear algebra components adapted to the particular structure of the KKT system. In case of the memory efficient algorithm, BELTISTOS, the memory usage is further reduced by releasing the memory required to store the L factors of the diagonal blocks, and recomputing the factorization during different phases of the algorithm. Obviously, the repeated factorizations in the memory efficient algorithm are reflected in increased execution time. However, the execution time is still orders of magnitude faster than general purpose approaches.

6.3. Complexity analysis

The factorization and backsolve phases, constituting the main building blocks of the direct sparse algorithms for the solution of KKT systems, are analyzed separately in this section, focusing on scaling with respect to the number of time periods N. For the MPOPF problems, the size of the KKT system linearly grows with the number of time periods, thus the complexity of the KKT solution algorithms

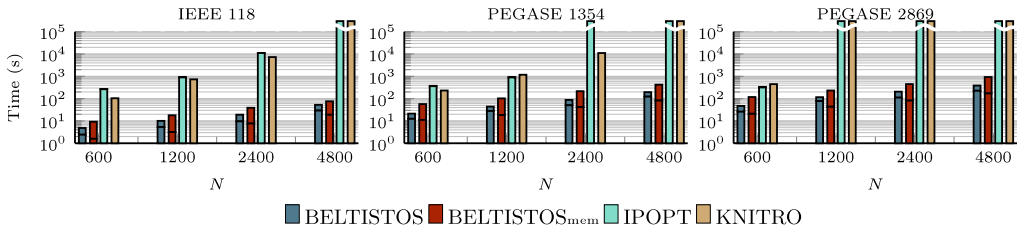


Fig. 6. Average numerical factorization and forward–backward substitutions time ($N_s = 10$).

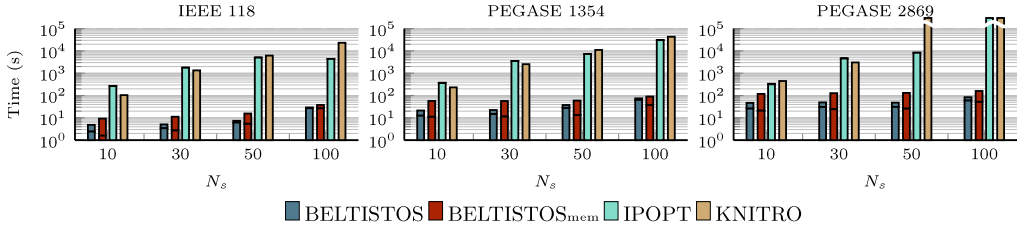


Fig. 7. Average numerical factorization and forward–backward substitutions time ($N = 600$).

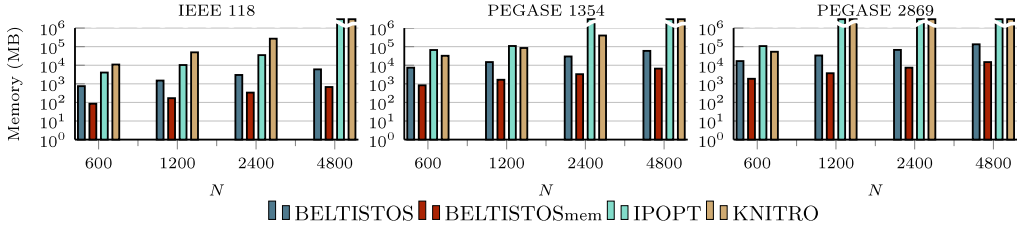


Fig. 8. Memory requirements for the optimizer ($N_s = 10$).

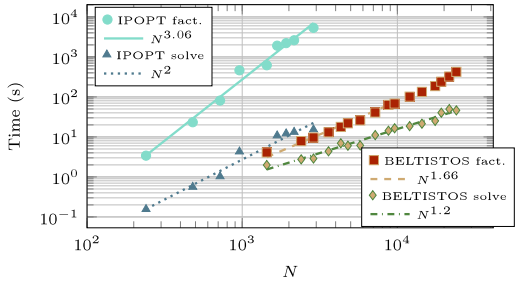


Fig. 9. Regression analysis for the factorization and solution phases using IEEE 118 bus benchmark.

needs to be analyzed in these terms. The complexity of BELTISTOS for solution of the MPOPF problems can be estimated by examining the complexity of the individual factorization and solution steps in Alg. 1, since they represent the most computationally expensive components of the algorithm. For the black box approach, represented by IPOPT, the factorization and solution phases are analyzed separately as well. Fig. 9 shows such complexity study of the factorization and solution phases. The tests are run using a standard power grid IEEE 118 bus benchmark in order to be able to demonstrate the algorithm also for larger number of time periods. As expected, the complexity for IPOPT is cubic for the factorization phase and quadratic for the solution phase. On the other hand, the scaling of BELTISTOS is almost by an order of magnitude better due to the decomposition of the KKT system and the block structure aware factorization. The figure also illustrates the fact that for increasing number of time periods, the performance gap between the standard and the structure exploiting approach becomes even more pronounced.

7. Concluding remarks

This work proposed a reliable IP optimization library BELTISTOS for solution of AC OPF problems. The computational and numerical performance was evaluated on four different AC OPF formulations, considering various aspects such as initializations, convergence tolerance and power grid networks of increasing complexity. It was demonstrated that BELTISTOS accelerates convergence of OPF problems by providing consistently accurate search directions regardless of OPF variable formulation, initialization, or network characteristics, even for severely ill-conditioned KKT systems. Similar performance for single period OPF problems was observed for KNITRO, while the performance of the other IP libraries considered in this study was inferior and varied based on the network complexity or the OPF formulation.

Furthermore, it was demonstrated that BELTISTOS achieves significant acceleration of the solution time via the aid of structure-exploiting algorithms for MPOPF problems. Additionally, these can also be configured for a significantly lower memory footprint. The solution times are orders of magnitude smaller compared to the best competitors, with the performance gap increasing for larger networks or longer time horizons. The adoption of these solution strategies, enables the solution of previously intractable time-coupled MPOPF problems, without relying on simplifications, or approximations, receding horizon approaches, or other decoupling techniques. In the future work, application cases spanning a full year horizon on hourly basis (with N =8760 time periods) will be investigated. This could enable transmission system operators to better plan their operations and reduce the risks arising from approximation errors.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.epsr.2022.108613.

References

[1] J. Carpentier, Contribution to the economic dispatch problem, Bull. Soc. Francoise Electr. 3 (8) (1962) 431–447.

[2] P. Lazzeroni, M. Repetto, Optimal planning of battery systems for power losses reduction in distribution grids, Electr. Power Syst. Res. 167 (2019) 94–112, http://dx.doi.org/10.1016/j.epsr.2018.10.027.

[3] M. Farrokhifar, F.H. Aghdam, A. Alahyari, A. Monavari, A. Safari, Optimal energy management and sizing of renewable energy and battery systems in residential sectors via a stochastic MILP model, Electr. Power Syst. Res. 187 (2020) 106483, http://dx.doi.org/10.1016/j.epsr.2020.106483.

[4] D. Kourounis, A. Fuchs, O. Schenk, Toward the next generation of multiperiod optimal power flow solvers, IEEE Trans. Power Syst. 33 (4) (2018) 4005–4014, http://dx.doi.org/10.1109/TPWRS.2017.2789187.

[5] T. Sayfutdinov, M. Ali, O. Khamisov, Alternating direction method of multipliers for the optimal siting, sizing, and technology selection of Li-ion battery storage, Electr. Power Syst. Res. 185 (2020) 106388, http://dx.doi.org/10.1016/j.epsr.2020.106388.

[6] J. Kardoš, D. Kourounis, O. Schenk, Structure-exploiting interior point methods, in: A. Grama, A.H. Sameh (Eds.), Parallel Algorithms in Computational Science and Engineering, Springer International Publishing, Cham, 2020, pp. 63–93, http://dx.doi.org/10.1007/978-3-030-43736-7_3.

[7] J. Kardoš, High-Performance Interior Point Methods: Application to Power Grid Problems (Ph.D. thesis), Università della Svizzera italiana, 2020, oai:doc.rero.ch:20200714102545-MX.

[8] S. Mehrotra, On the implementation of a primal-dual interior point method, SIAM J. Optim. 2 (4) (1992) 575–601, http://dx.doi.org/10.1137/0802028, arXiv:https://doi.org/10.1137/0802028.

[9] A. Garzillo, M. Innorta, M. Ricci, The problem of the active and reactive optimum power dispatching solved by utilizing a primal-dual interior point method, Int. J. Electr. Power Energy Syst. 20 (6) (1998) 427–434, http://dx.doi.org/10.1016/S0142-0615(98)00010-6.

[10] G.L. Torres, V.H. Quintana, An interior-point method for nonlinear optimal power flow using voltage rectangular coordinates, IEEE Trans. Power Syst. 13 (4) (1998) 1211–1218, http://dx.doi.org/10.1109/59.736231.

[11] F. Capitanescu, M. Glavic, D. Ernst, L. Wehenkel, Interior-point based algorithms for the solution of optimal power flow problems, Electr. Power Syst. Res. 77 (5) (2007) 508–517, http://dx.doi.org/10.1016/j.epsr.2006.05.003.

[12] F. Capitanescu, L. Wehenkel, Experiments with the interior-point method for solving large scale Optimal Power Flow problems, Electr. Power Syst. Res. 95 (2013) 276–283, http://dx.doi.org/10.1016/j.epsr.2012.10.001.

[13] Z. Yuan, M. Paolone, Properties of convex optimal power flow model based on power loss relaxation, Electr. Power Syst. Res. 186 (2020) 106414, http://dx.doi.org/10.1016/j.epsr.2020.106414.

[14] B. Park, C.L. DeMarco, Convex relaxation of Sparse Tableau Formulation for the AC optimal power flow, Electr. Power Syst. Res. 171 (2019) 209–218, http://dx.doi.org/10.1016/j.epsr.2019.02.020.

[15] M. Ma, L. Fan, Z. Miao, B. Zeng, H. Ghassempour, A sparse convex AC OPF solver and convex iteration implementation based on 3-node cycles, Electr. Power Syst. Res. 180 (2020) 106169, http://dx.doi.org/10.1016/j.epsr.2019.106169.

[16] C. Coffrin, R. Bent, K. Sundar, Y. Ng, M. Lubin, PowerModels.Jl: An open-source framework for exploring power flow formulations, in: 2018 Power Systems Computation Conference, PSCC, IEEE, 2018, pp. 1–8, http://dx.doi.org/10.23919/PSCC.2018.8442948.

[17] A. Castillo, R.P. O'Neill, Computational Performance of Solution Techniques Applied to the ACOPF, Tech. Rep., Federal Energy Regulatory Commission, 2013, URL https://www.ferc.gov/sites/default/files/2020-05/acopf-5-computational-testing.pdf.

[18] N. Gould, D. Orban, A. Sartenaer, P.L. Toint, Superlinear convergence of primal-dual interior point algorithms for nonlinear programming, SIAM J. Optim. 11 (4) (2001) 974–1002, http://dx.doi.org/10.1137/S1052623400370515.

[19] D. Kourounis, J. Kardos, O. Schenk, BELTISTOS – A robust interior point optimizer for large-scale optimal power flow problems, 2020, www.beltistos.com.

[20] R.D. Zimmerman, C.E. Murillo-Sanchez, R.J. Thomas, MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education, IEEE Trans. Power Syst. 26 (2011) 12–19, http://dx.doi.org/10.1109/TPWRS.2010.2051168.

[21] R.D. Zimmerman, C.E. Murillo-Sanchez, MATPOWER 7.0 User's Manual, 2019, http://dx.doi.org/10.5281/zenodo.3251118.

[22] J. Kardos, D. Kourounis, O. Schenk, R.D. Zimmerman, Complete results for a numerical evaluation of interior point solvers for large-scale optimal power flow problems, 2020, arXiv:1807.03964.

[23] E. Stai, F. Sossan, E. Namor, J.-Y.L. Boudec, M. Paolone, A receding horizon control approach for re-dispatching stochastic heterogeneous resources accounting for grid and battery losses, Electr. Power Syst. Res. 185 (2020) 106340, http://dx.doi.org/10.1016/j.epsr.2020.106340.

[24] D. Kourounis, O. Schenk, Method to accelerate the processing of multiperiod optimal power flow problems, 2018, European Patent Nr. EP 3602325, USA Patent Nr: US2020/0042569. https://patents.google.com/patent/WO2018177529A1.

[25] A. Wächter, L.T. Biegler, Line search filter methods for nonlinear programming: motivation and global convergence, SIAM J. Optim. 16 (1) (2005) 1–31, http://dx.doi.org/10.1137/S1052623403426556 (electronic).

[26] A. Wächter, L.T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Math. Program. 106 (1, Ser. A) (2006) 25–57, http://dx.doi.org/10.1007/s10107-004-0559-y.

[27] J. Nocedal, S.J. Wright, Numerical Optimization, second ed., Springer, New York, NY, USA, 2006.

[28] O. Schenk, K. Gärtner, On fast factorization pivoting methods for sparse symmetric indefinite systems, Electron. Trans. Numer. Anal. 23 (2006) 158–179.

[29] I.S. Duff, MA57 - A New Code for the Solution of Sparse Symmetric Definite and Indefinite Systems, Tech. Rep. RAL-TR-2002-024, Rutherford Appleton Laboratory, 2002, http://www.hsl.rl.ac.uk.

[30] M. Bollhöfer, O. Schenk, R. Janalik, S. Hamm, K. Gullapalli, State-of-the-art sparse direct solvers, in: A. Grama, A.H. Sameh (Eds.), Parallel Algorithms in Computational Science and Engineering, Springer International Publishing, Cham, 2020, pp. 3–33, http://dx.doi.org/10.1007/978-3-030-43736-7_1.

[31] O. Schenk, M. Bollhöfer, R.A. Römer, On large-scale diagonalization techniques for the Anderson model of localization, SIAM Rev. 50 (1) (2008) 91–112, http://dx.doi.org/10.1137/070707002.

[32] M. Bollhöfer, M.J. Grote, O. Schenk, Algebraic multilevel preconditioner for the Helmholtz equation in heterogeneous media, SIAM J. Sci. Comput. 31 (5) (2009) 3781–3805, http://dx.doi.org/10.1137/080725702.

[33] M. Bollhöfer, Y. Saad, Multilevel preconditioners constructed from inverse-based ILUs, SIAM J. Sci. Comput. 27 (5) (2006) 1627–1650, http://dx.doi.org/10.1137/040608374.

[34] I. Corporation, Developer Reference for Intel Math Kernel Library, Intel Corporation, 2019, Chapter 5: Sparse Solver Routines. URL https://software.intel.com/en-us/download/developer-reference-for-intel-math-kernel-library-c.

[35] U.M. Ascher, C. Greif, A First Course in Numerical Methods, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2011, http://dx.doi.org/10.1137/9780898719987.

[36] Y. Saad, Iterative Methods for Sparse Linear Systems, second ed., Society for Industrial and Applied Mathematics, 2003, http://dx.doi.org/10.1137/1.9780898718003.

[37] M. Bollhöfer, O. Schenk, F. Verbosio, A high performance level-block approximate LU factorization preconditioner algorithm, Appl. Numer. Math. 162 (2021) 265–282, http://dx.doi.org/10.1016/j.apnum.2020.12.023.

[38] J. Kardoš, D. Kourounis, O. Schenk, Two-level parallel augmented schur complement interior-point algorithms for the solution of security constrained optimal power flow problems, IEEE Trans. Power Syst. 35 (2) (2020) 1340–1350, http://dx.doi.org/10.1109/TPWRS.2019.2942964.

[39] R.H. Byrd, J. Nocedal, R.A. Waltz, Knitro: An integrated package for nonlinear optimization, in: Large-Scale Nonlinear Optimization, Springer US, Boston, MA, 2006, pp. 35–59, http://dx.doi.org/10.1007/0-387-30065-1_4.

[40] H. Wang, C.E. Murillo-Sanchez, R.D. Zimmerman, R.J. Thomas, On computational issues of market-based optimal power flow, IEEE Trans. Power Syst. 22 (3) (2007) 1185–1193, http://dx.doi.org/10.1109/TPWRS.2007.901301.

[41] R.D. Zimmerman, H. Wang, MATPOWER Interior Point Solver (MIPS) User's Manual, Zenodo, 2019, http://dx.doi.org/10.5281/zenodo.3251014.

[42] R.H. Byrd, J.C. Gilbert, J. Nocedal, A trust region method based on interior point techniques for nonlinear programming, Math. Program. 89 (1) (2000) 149–185, http://dx.doi.org/10.1007/PL00011391.

[43] S. Fliscounakis, P. Panciatici, F. Capitanescu, L. Wehenkel, Contingency ranking with respect to overloads in very large power systems taking into account uncertainty, preventive, and corrective actions, IEEE Trans. Power Syst. 28 (4) (2013) 4909–4917, http://dx.doi.org/10.1109/TPWRS.2013.2251015.

[44] C. Josz, S. Fliscounakis, J. Maeght, P. Panciatici, AC power flow data in MATPOWER and QCQP format: itesla, RTE snapshots, and PEGASE, 2016, arXiv:1603.01533.

[45] A.B. Birchfield, T. Xu, K.M. Gegner, K.S. Shetye, T.J. Overbye, Grid structural characteristics as validation criteria for synthetic networks, IEEE Trans. Power Syst. 32 (4) (2017) 3258–3265, http://dx.doi.org/10.1109/TPWRS.2016.2616385.

[46] A. Castillo, R.P. O'Neill, Survey of Approaches to Solving the ACOPF, Tech. Rep., Federal Energy Regulatory Commission, 2013, URL https://www.ferc.gov/sites/default/files/2020-05/acopf-4-solution-techniques-survey.pdf.

[47] E. Dolan, J. Moré, Benchmarking optimization software with performance profiles, Math. Program. 91 (2002) 201–213, http://dx.doi.org/10.1007/s101070100263.

[48] N. Gould, J.A. Scott, A numerical evaluation of HSL packages for the direct solution of large sparse, symmetric lin. Systems of equations, ACM Trans. Math. Softw. 30 (3) (2004) 300–325, http://dx.doi.org/10.1145/1024074.1024077.