Large-Scale Security-Constrained Optimal Power Flow Problems with Energy Storage Systems

Nico Meyer-Huebner, Student Member, IEEE, Abolfazl Mosaddegh, Student Member, IEEE, Michael Suriyah, Member, IEEE, Thomas Leibfried, Member, IEEE, Claudio A. Cañizares, Fellow, IEEE, and Kankar Bhattacharya, Fellow, IEEE

Abstract—In this paper, a novel approach is presented to solve large-scale Security-Constrained Optimal Power Flow (SCOPF) problems associated with power grids containing Energy Storage Systems (ESSs), using a distributed computing architecture, referred to as a Smart Grid Communication Middleware (SGCM) system. The problem is solved by splitting the time horizon into shorter time intervals using the SGCM system, solving one iteration of a Primal Dual Interior Point Method (PDIPM) for each resulting sub-problem in parallel and exchanging boundary variables after each iteration in three different stages. The sub-problems eventually converge to a solution near to an optimal solution of the entire problem without having to solve the original SCOPF Karush-Kuhn-Tucker (KKT)-conditions. The methodology is tested on the German transmission grid, where the computational effort is shown to be reduced significantly; thus, with an entire horizon of 96 time steps and a decomposition into 8 sub-problems, the run-time can be decreased from over 2 hours to less than 10 minutes with an overall system cost increase of less than 1%, which makes the SCOPF model with inter-temporal ESS constraints suitable for online applications.

Keywords—Distributed computing, energy storage, large-scale power grids, optimal power flow, optimization, smart grids.

NOMENCLATURE

Indices

\( n \) Number of inequalities.

\( N \) Number of sub-problems.

Functions

\( \tilde{\gamma}_{\text{lin}}, \tilde{\gamma}_{\text{aln}} \) Time-independent equality constraints.

\( \tilde{\gamma}_{\text{lin}}, \tilde{\gamma}_{\text{aln}} \) Time-dependent inequality constraints.

\( \tilde{\gamma}_{\text{lin}}, \tilde{\gamma}_{\text{aln}} \) Lagrangian function.

Parameters

\( E_{S_{\text{min}}}, E_{S_{\text{max}}} \) Minimum and maximum storage energy [p.u.].

\( n_{i} \) Number of inequalities.

\( N \) Number of sub-problems.

Variables

\( L \) Linearly constrained parts of \( x \) [p.u.].

\( L_{\text{min}}, L_{\text{max}} \) Minimum and maximum limits of \( L \) [p.u.].

\( E_{S} \) Storage energy [p.u.].

\( P_{\text{bus}}, Q_{\text{bus}} \) Injected active and reactive powers [p.u.].

\( P_{g}, P_{g} \) Active power output of generators [p.u.].

\( P_{g}, P_{g} \) Lower and upper limit of generator powers at boundaries [p.u.].

\( P_{S_{c}}, P_{S_{d}} \) Fixed generator powers and storage energy values [p.u.].

\( S_{\text{br}}, P_{S_{c}}, P_{S_{d}} \) Storage power outputs for charging, discharging [p.u.].

\( S_{\text{br}}, P_{S_{c}}, P_{S_{d}} \) Apparent branch flow magnitudes [p.u.].

\( V \) Rectangular-coordinate node-voltage diagonal matrix [p.u.].

\( x \) All optimization variables.

\( Z \) Slack variables.

\( \hat{\alpha}, \hat{\alpha}_{0} \) Absolute feasibility condition and initial feasibility condition.

\( \alpha \) Relative feasibility condition.

\( \gamma \) PDIPM perturbation parameter.

\( \Delta E_{S} \) Incremental energy in storage systems when charging or discharging [p.u.].

\( \hat{\alpha}, \hat{\gamma}, \hat{\delta}_{E}, \hat{\delta}_{\text{Up}}, \hat{\delta}_{\text{Down}}, \hat{\delta}_{p}, \hat{\delta}_{\text{Up}}, \hat{\delta}_{\text{Down}}, \hat{\delta}_{p} \) Absolute gaps of storage energy at boundaries [p.u.].

\( \hat{\delta}_{E}, \hat{\delta}_{p} \) Relative maximal gap of storage energy and generator ramping at boundaries [p.u.].

\( \lambda, \mu \) Lagrangian multipliers.

N. Meyer-Huebner, M. Suriyah and T. Leibfried (\{nico.meyer-huebner, michael.suriyah, thomas.leibfried\}@kit.edu) are with Karlsruhe Institute of Technology, Karlsruhe, Germany.

A. Mosaddegh, C. A. Cañizares, and K. Bhattacharya (\{amosaddegh, ccانيزاردز, kankar\}@uwaterloo.ca) are with University of Waterloo, Waterloo, ON, Canada.
I. INTRODUCTION

THE Optimal Power Flow (OPF) problem was introduced in the sixties [1], and has ever since been widely used in many power system applications [2], [3]. The classical OPF problem involves finding the optimal generation schedule to minimize operating costs while respecting network constraints such as voltage and line capacity limits. On the other hand, secure network operation is becoming more challenging with the increasing penetration of Renewable Energy Sources (RESs), since the fluctuating output from these RESs requires more flexibility in generator ramping capacity, as well as availability of Energy Storage Systems (ESSs).

In order to properly take into account RES and ESS restrictions, it is necessary to include inter-temporal/time-constraint constraints in the OPF model, such as standard generator ramp-up and ramp-down as well as ESS charging/discharging ramping and limits in the OPF models, resulting in a Security-Constrained OPF (SCOPF) problem, based on RES and load profile forecasts. In [4], efficient computation methods for the SCOPF problem are presented using Interior Point Methods (IPM). More recently, OPFs for active network management in distribution grids with ESS have been proposed [5], [6]. Thus, in [5], a framework is proposed to optimize energy storage as well as flexible demand in a distribution grid, and in [6], the OPF is embedded in an architecture to cope with forecast uncertainty to optimally dispatch generators and storage units. In these works, the network sizes are usually not too large; however, when considering inter-connected transmission grids, the problem size grows significantly with increasing time horizon and/or grid size. A single OPF in a large-scale network is already challenging, since it is a nonlinear and non-convex problem; if multiple time steps are considered, the problem size grows exponentially and the solution times do not fulfill requirements of online applications. Thus, there is a need to reduce the computational effort of large-scale SCOPF problems with ESS.

Some modeling approaches can reduce the number of non-zero entries in the Karush-Kuhn-Tucker (KKT) conditions and thus reduce the computational effort [7]. A multi-area decomposition methodology is presented and applied to conventional OPF problems in [8], and the same methodology is adopted to solve the SCOPF problem in [9]. A different approach to the multi-area SCOPF is shown in [10], where the areas are solved using available storage capacities of external areas by means of a game theory approach. A direct solution method for large-scale SCOPF problems is proposed in [11], where a constant diagonal perturbation is introduced in the KKT conditions to utilize Cholesky factorization, and reordering algorithms are integrated in the solver to improve efficiency. The block structure of KKT conditions is exploited through decomposition in [12] and [13], where for every iteration, the global KKT-system is partitioned into sub-blocks, which are solved independently with a direct solver in [12], or an iterative solver in [13]. The decomposition of all aforementioned works is either based on spatial areas, or require fundamental low level solver programming. The present paper proposes a different approach based on a time scale decomposition, while keeping the implementation simple to be able to use standard open-source solvers.

In [8], a multi-area decomposition approach is applied to a single time step OPF; the sub-problems exchange their boundary variables after each Newton step. Similar to [8], the current paper proposes a novel decomposition approach applied to the desired time horizon. The time-dependent variables between the sub-problems are exchanged after each iteration of the Primal-Dual Interior Point Method (PDIPM), as proposed in [14]. The variable exchange process is divided into three stages where boundary conditions are treated initially as loose and then sequentially tightened, with the PDIPM taking longer to solve as the boundary conditions are firmed up. To take advantage of the independence between the sub-problems, the Smart Grid Communication Middleware (SGCM) system is chosen as the preferred platform, since it has proven to be an efficient distributed computing architecture in [15], [16], and is based on distributing the sub-problems among computing worker-nodes, which calculate each sub-problem in parallel. Therefore, the main contributions of the present paper are as follows:

- Develop a novel decomposition methodology for SCOPF problems with ESS by decoupling inter-temporal constraints.
- Implement the distributed computation of the decomposed problem using the SGCM system.
- Demonstrate the efficiency of the proposed approach for possible online realistic applications using the 1215-bus German transmission grid.

The rest of the paper is structured as follows: In Section II, the general SCOPF problem with ESS is presented, and the PDIPM solution technique is briefly described. In Section III, the proposed decomposition methodology is described in detail, and the distributed computing architecture used for solving the decomposed sub-problems is briefly described. The results for a realistic case study based on the German transmission grid are presented and discussed in Section IV, and the main conclusions of the presented work are provided in Section V.

II. MATHEMATICAL MODEL OF SCOPF

A. Static Constraints

The conventional OPF constraints for each time step \( t \) include the nonlinear load flow equations:

\[
P_{\text{bus},t} - \text{Re}(V_t Y_t V_t^*) = 0 \tag{1}
\]

\[
Q_{\text{bus},t} - \text{Im}(V_t Y_t V_t^*) = 0 \tag{2}
\]

In this work, voltage is expressed in rectangular coordinates and the voltage limits are thus given as quadratic functions:

\[
V_{\text{min}}^2 \leq |V_{\text{ref}}|^2 \leq V_{\text{max}}^2 \quad \forall m \tag{3}
\]

The branch flow apparent power is limited by:

\[
S_{\text{br}} \leq S_{\text{br,max}} \tag{4}
\]

A reference angle must be provided, which is achieved by setting the imaginary component of the voltage of one of
where the state vector can be defined as:

\[ L_{\text{min}} \leq L_t \leq L_{\text{max}}, \]

Note that some limits change from time step to time step, as these include fluctuating RESs or load profiles. All variables in these and other equations are defined in the Nomenclature section.

**B. Dynamic Constraints**

In addition to the conventional OPF constraints, there are time-dependent constraints when multiple time steps are considered simultaneously. These include ramp rates of generators and operating constraints of ESSs, as explained next.

1) Generator Ramp Rates: The power output of generators depends on the generated power from one time step earlier. The ramp rates \( R_{\text{up}} \) and \( R_{\text{down}} \) denote the active power ramp-up and -down limits, respectively, as follows:

\[ P_{g t} - P_{g t-1} \leq R_{\text{up}} \]
\[ P_{g t-1} - P_{g t} \leq R_{\text{down}} \]

2) ESS: ESSs are considered technology-independent in this work. The incremental energy stored in the energy storage during an incremental time \( \Delta t \) is given by:

\[ \Delta E_{s k,t} = (\eta_{k,t} P_{\text{sc},k,t} - \eta_{d,k,t} P_{\text{sd},k,t}) \Delta t \quad \forall k \]

where:

\[ 0 \leq P_{\text{sc}} \leq P_{\text{sc,max}} \]
\[ 0 \leq P_{\text{sd}} \leq P_{\text{sd,max}} \]

The stored energy at time \( t \) is then calculated as:

\[ E_{s t} = E_{s t-1} + \Delta E_{s t} \]

with:

\[ E_{s \text{min}} \leq E_{s t} \leq E_{s \text{max}} \]

**C. SCOPF**

A general SCOPF problem can be described as follows:

\[ \min_{x} \quad F(x) \]
\[ \text{s.t.} \quad g(x) = 0 \]
\[ h(x) \leq 0 \]

where a cost function \( F(x) \) is minimized in (13), while respecting the equality constraints \( g(x) \) in (14), and the inequality constraints \( h(x) \) in (15). In the SCOPF formulation, the state vector can be defined as:

\[ x = \begin{bmatrix} x_1 \\ \vdots \\ x_t \\ \vdots \\ x_T \end{bmatrix} \]

where \( x_t \) includes the optimization variables, such as voltages, generator outputs, storage power and energy, etc., for time step \( t \). Constraints (14) and (15) can be split into two types of constraints: nonlinear constraints \( g_{\text{lin}}(x) \) and \( h_{\text{lin}}(x) \), and linear constraints \( g_{\text{lin}}(x) \) and \( h_{\text{lin}}(x) \):

\[ g(x) = \begin{bmatrix} g_{\text{lin}}(x) \\ g_{\text{nonlin}}(x) \end{bmatrix} \quad h(x) = \begin{bmatrix} h_{\text{lin}}(x) \\ h_{\text{nonlin}}(x) \end{bmatrix} \]

While the nonlinear constraints are entirely static, the linear constraints are partly time-dependent. Constraints \( g_{\text{lin}}(x) \) include the definition of the reference angle, and time-dependent constraints \( \bar{g}_{\text{lin}}(x) \) associated with (11); \( h_{\text{lin}}(x) \) includes static inequalities (5), (9), (10), and (12), and time-dependent constraints \( \bar{h}_{\text{lin}}(x) \) contain (6) and (7). Thus:

\[ g_{\text{lin}}(x) = \begin{bmatrix} \tilde{g}_{\text{lin}}(x_1) \\ \tilde{g}_{\text{lin}}(x_t) \\ \vdots \\ \tilde{g}_{\text{lin}}(x_T) \end{bmatrix} \quad h_{\text{lin}}(x) = \begin{bmatrix} \tilde{h}_{\text{lin}}(x_1) \\ \tilde{h}_{\text{lin}}(x_t) \\ \vdots \\ \tilde{h}_{\text{lin}}(x_T) \end{bmatrix} \]

The state vector \( x_0 \) describes the initial state for inter-temporal constraints like the ESS State of Charge (SOC) or the power output of generators with ramping limits. The total cost is thus given as:

\[ F(x) = \sum_{t=1}^{T} F(x_t) \]

where the cost function of a generator can be a piecewise-linear or polynomial function.

**D. Solution [14]**

The SCOPF problem is solved using a PDIPM, assigning Lagrangian multipliers \( \lambda \) to the equality constraints and \( \mu \) to the inequality constraints. To include the inequalities in a function to be minimized, slack variables \( Z > 0 \) are introduced, so that the inequalities are transformed into equality constraints \( h(x) + Z = 0 \), resulting in the following Lagrangian for a given \( \gamma \):

\[ \mathcal{L}(x, \lambda, \mu, Z) = F(x) + \mu^T (h(x) + Z) + \lambda^T g(x) - \gamma \sum_{j=1}^{n} \ln(Z_j) \]

The slack variables are weighted with the barrier coefficient \( \gamma \) to keep the inequality constraints away from zero in the first
iterations. Eventually, $\gamma$ is reduced to zero so that the problem approaches the original one. The Lagrangian is minimized, while maximizing the multipliers $\lambda$ and $\mu$. This leads to a minimization of the cost function $F$, while $h(x)$ is kept below zero and $g(x)$ is close to zero.

Common Newton steps are applied to solve the KKT-conditions for (21) [17]. The perturbation parameter $\gamma$ must converge to zero in order to satisfy the first order optimality conditions of the original problem, and is updated with the rule:

$$ \gamma \leftarrow \sigma \frac{Z^T \mu}{n_i}, $$

where $n_i$ is the number of inequalities and $\sigma$ is a scalar constant between 0.1 and 0.3 in this paper; this value indicates the speed at which $\gamma$ decreases, and therefore, the speed at which the original problem is approached. Typically, $\sigma = 0.1$ is a good trade-off between convergence speed and robustness.

III. DECOMPOSITION METHODOLOGY

In Fig. 1, the flowchart of the proposed methodology is shown. First, the problem is split into sub-problems and each sub-problem is initialized for the PDIPM. Then, the sub-problems are distributed among the worker-nodes of the SGCM system [15], [16]. Each worker-node performs one iteration of the PDIPM, including an update for the new search direction. The update is saved in the worker-node and the boundary variables, i.e. storage energy levels and power output of ramping constrained generators at the beginning and at the end of the sub-problem, are returned to the master-node. The master-node checks the progress of the optimization and the boundary variables are exchanged or modified depending on the stage of the solution process. The sub-problems with the new boundary conditions are distributed again amongst worker-nodes, and the next iteration starts. Once all sub-problems have converged, the master-node collects and outputs the final results.

A. Decomposition

If the original problem has $T$ time steps, $T$ must be divisible by the number of sub-problems $N$. Therefore, $T = NT_p$, where the horizon of one sub-problem, i.e. the number of time steps, is $T_p$. In Fig. 2, the decomposition of the solution vector $x$ is shown; the superscripts denote the number of the sub-problem. Thus, the linear and nonlinear constraints in (18) and (19) are decomposed into $N$ new sets of constraints, as follows:

$$ g_{\text{lin}}(x^1) = \begin{bmatrix} g_{\text{lin}}(x_1) \\ \vdots \\ g_{\text{lin}}(x_{T_p}) \end{bmatrix} $$

(23)

$$ g_{\text{lin}}(x^p) = \begin{bmatrix} g_{\text{lin}}(x_{(p-1)T_p+1}) \\ \vdots \\ g_{\text{lin}}(x_{pT_p}) \end{bmatrix} $$

(24)

$$ h_{\text{lin}}(x^1) = \begin{bmatrix} h_{\text{lin}}(x_1) \\ \vdots \\ h_{\text{lin}}(x_{T_p}) \end{bmatrix} $$

(25)

$$ h_{\text{lin}}(x^p) = \begin{bmatrix} h_{\text{lin}}(x_{(p-1)T_p+1}) \\ \vdots \\ h_{\text{lin}}(x_{pT_p}) \end{bmatrix} $$

(26)

$$ h_{\text{lin}}(x^N) = \begin{bmatrix} h_{\text{lin}}(x_{(N-1)T_p+1}) \\ \vdots \\ h_{\text{lin}}(x_{T_p}) \end{bmatrix} $$

(27)

$$ h_{\text{lin}}(x^{N-1}) = \begin{bmatrix} h_{\text{lin}}(x_T) \\ \vdots \\ h_{\text{lin}}(x_{(N-1)T_p+1}) \end{bmatrix} $$

(28)
Fig. 2. Decomposition of the original problem into N sub-problems with \( N - 1 \) boundaries.

\[
\begin{align*}
\mathbf{x}^1 &= \mathbf{x}_1 \\
\vdots \\
\mathbf{x}^p &= \mathbf{x}_p \\
\vdots \\
\mathbf{x}^N &= \mathbf{x}_N \\
\mathbf{bnd}_1 \\
\vdots \\
\mathbf{bnd}_{p-1} \\
\mathbf{bnd}_p \\
\vdots \\
\mathbf{bnd}_{N-1} \\
\mathbf{bnd}_p \\
\vdots \\
\mathbf{bnd}_{N-1} \\
\mathbf{bnd}_p \\
\vdots \\
\mathbf{bnd}_1 \\
\end{align*}
\]

\[
\mathbf{h}_{\text{lin}}(\mathbf{x}) = \begin{bmatrix}
\mathbf{h}_{\text{lin}}(x_1) \\
\vdots \\
\mathbf{h}_{\text{lin}}(x_{TP}) \\
\mathbf{h}_{\text{lin}}(x_{TP-1}, x_{TP}) \\
\end{bmatrix}
\]

B. Stages

The boundaries amongst the sub-problem blocks in Fig. 2 are defined by the ramp and storage constraints (6)-(8) and (11). It is proposed here that these boundaries are treated in three different stages of convergence, with decreasing level of ramp constraint relaxation, from loose to tight, as explained next.

1) Stage I: The boundary constraints are completely relaxed, i.e. are ignored, and the sub-problems are therefore decoupled from each other. This enables them to evolve into a direction that is favorable for each specific sub-problem, decreasing the dependency on the starting point. The PDIPM \( \sigma \) value is set to 0.3, in order not to converge too quickly to a solution. The resulting feasibility condition \( \alpha_p \) can then be determined for each sub-problem \( p \) as follows:

\[
\alpha_p = \frac{\max(\|h(x_p)\|_\infty, \|g(x_p)\|_\infty)}{1 + \max(\|\lambda_p\|_\infty, \|\mu_p\|_\infty)}
\]

(35)

(36)

The variable \( \alpha_p \) is calculated at every iteration, while \( \alpha_{0,p} \) is only calculated once at the start, and thus depends on the starting point.

2) Stage II: Once the feasibility condition has evolved to \( \alpha_p < 0.1 \) for all \( N \) sub-problems, the boundary conditions are then enforced, with a \( \sigma \) reduced to 0.2. For a boundary between sub-problems \( p \) and \( p+1 \), the sub-problem \( p \) considers the following constraints for the next iteration:

\[
E_{s_1}^{p+1} - \Delta E_{s_1}^{p+1} \leq E_{s_{p+1}}^{p} \leq E_{s_1}^{p+1} - \Delta E_{s_1}^{p+1} \quad (37)
\]

\[
P_g^{p+1} \leq P_g^{p}_{p+1} \leq P_g^{p}_{T_p} \quad (38)
\]

(37)

(38)
where

\[ P_{g_{l,T_p}} = \max \left( \min \left( P_{g_{l,T_p}^{p+1}}, R_{l}^{Up}, P_{g_{\max}} \right), P_{g_{\min}} \right) \forall l \]  

\[ P_{g_{l,T_p}^{p}} = \min \left( \max \left( P_{g_{l,T_p}^{p+1}}, R_{l}^{Up}, P_{g_{\max}} \right), P_{g_{\min}} \right) \forall l \]  

The sub-problem \( p + 1 \) considers the following constraints:

\[ E_{s_{l,T_p}^{p}} \leq E_{s_{l,T_p}^{p+1}} - \Delta E_{s_{l,T_p}^{p+1}} \leq E_{s_{l,T_p}^{p+1}} \]  

\[ P_{g_{l,T_p}^{p+1}} \leq P_{g_{l,T_p}^{p+1}} \leq P_{g_{l,T_p}^{p+1}} \]  

where

\[ P_{g_{l,T_p}^{p+1}} = \max \left( \min \left( P_{g_{l,T_p}^{p+1}}, R_{l}^{Up}, P_{g_{\max}} \right), P_{g_{\min}} \right) \forall l \]  

\[ P_{g_{l,T_p}^{p+1}} = \min \left( \max \left( P_{g_{l,T_p}^{p+1}}, R_{l}^{Up}, P_{g_{\max}} \right), P_{g_{\min}} \right) \forall l \]  

Note that (37) and (41) are in principle equality constraints; however, these are considered as inequality constraints at this stage to treat them as relaxed constraints, thus reducing computational burden. Observe also, that (37), (38), and (42) are part of the model and only need a change of limits, whereas (41) needs to be added to the model explicitly. For simplicity, the latter is included from the beginning, but at Stage I, the limits are set to the standard maximum and minimum energy values for the ESSs.

The gap between the time-dependent variables on both sides of the boundary \( bnd_p \) can be calculated as follows from the solution of this stage for Stage III:

\[ \delta_{bnd} = \left| E_{s_{l,T_p}^{p+1}} - E_{s_{l,T_p}^{p+1}} - \Delta E_{s_{l,T_p}^{p+1}} \right| \forall k \]  

\[ \delta_{Up} = P_{g_{l,T_p}^{p+1}} - P_{g_{l,T_p}^{p+1}} - R_{l}^{Up} \]  

\[ \delta_{Down} = P_{g_{l,T_p}^{p+1}} - P_{g_{l,T_p}^{p+1}} - R_{l}^{Down} \]  

The convergence criteria at the boundaries are chosen as:

\[ \delta_{bnd} = \max \left( \frac{\delta_{Up}}{P_{g_{l,T_p}^{p+1}}}, \frac{\delta_{Down}}{P_{g_{l,T_p}^{p+1}}} \right) \forall l \]  

\[ \delta_{bnd} = \max \left( \frac{\delta_{bnd}}{P_{g_{l,T_p}^{p+1}}} \right) \forall k \]  

If the values of \( \delta_{bnd} \) and \( \delta_{bnd} \) are below 0.03, the boundaries can be considered fixed, i.e., converged; this means that a deviation of 3% between two sub-problems is considered as a satisfied boundary condition. This tolerance is chosen considering the trade-off between the algorithm ease of convergence (larger value) and optimality (smaller value).

The mean value between the two sub-problems is then calculated, and the remaining of the generator ramping on both sides of the boundary is shared equally between the two sub-problems as follows:

\[ E_{s_{bnd}} = \max \left( \frac{1}{2} \left( E_{s_{k,T_p}^{p+1}} + E_{s_{k,T_p}^{p+1}} - \Delta E_{s_{k,T_p}^{p+1}} \right), 0 \right) \forall k \]  

3) Stage III: From (50)-(55), the boundary constraints are defined as follows for the last stage:

\[ E_{s_{bnd}} \leq E_{s_{bnd}} \leq E_{s_{bnd}} \]  

\[ E_{bnd} \leq E_{bnd} \leq E_{bnd} \]  

\[ P_{g_{bnd}} \leq P_{g_{bnd}} \leq P_{g_{bnd}} \]  

\[ P_{g_{bnd}} \leq P_{g_{bnd}} \leq P_{g_{bnd}} \]  

Here, \( \sigma \) is set to the standard OPF value of 0.1. Then, each worker can compute the final solution of the respective sub-problem independently, and after all sub-problems have converged, the master-node yields the final solution, without the need for further iterations between the worker-nodes and the master-node.

C. Distributed Computing Architecture

The distributed computing architecture based on the SGCM system, as explained in some detail in [15], [16], is used here to reduce the run-time of the SCOPF model. The SGCM system, which is a framework for a “MapReduce” model, is a platform for parallel computing. The proprietary protocols on top of Transmission Control Protocol (TCP)/Internet Protocol (IP) for inter-communication of computing components are used in this system, while its core components include a console and nodes. The console is a client program that is responsible for application deployment, installation of master and worker applications, execution, and management of the model. The computing nodes include the worker-nodes and the master-node. These nodes are able to make TCP connections to other nodes, and also wait for TCP connections to provide services. These nodes are responsible to execute groups of instructions, deploying, calling, bundling, transferring, executing, and managing files.

The main steps of the MapReduce model are “Map” and “Reduce.” The master-node divides the data into smaller sub-problems based on the work-flow logic of the algorithm, and maps those sub-problems to related worker-nodes. When the worker-nodes finish their processing, the master-node collects the results from all worker-nodes, analyzes the data based on the algorithm, and reduces/combines the data into an output format. Since each worker-node operates independently, the problem is simultaneously run as different sub-problems,
reducing the computational cost considerably. If some modifications are needed after the Map operation, a “Shuffle” mid-stage is introduced in the MapReduce model to sort or exchange data between worker-nodes, and finally, prepare the data for the Reduce operation. Some worker-nodes may act as master-nodes for their sub-problems in a tree structure.

The proposed distributed computing approach applied to the SCOPF problem was executed on a server based on the Windows 64-bit operating system, with two E5-2670 v3 Intel Xeon 2.30 GHz processors and 128 GB RAM with 48 cores. The computing nodes with different port addresses and the console were all installed in this server.

IV. RESULTS

The methodology is tested on a realistic 1215-bus model of the German transmission grid with 96 time steps of 1 hour, i.e., for a 4-day horizon. A more detailed description of the network model can be found in [18]. First, the results of the presented methodology are compared with the solution of the original integrated problem; then, the computational effort and specific parameters concerning the decomposition methodology are discussed.

A. Solution Comparison

To compare the solutions of the decomposed problems with the original problem, the time coupling of the ESSs are considered. In Fig. 3(a), the accumulated energy of all ESSs in the grid is shown for the original problem and for a decomposition into 4 sub-problems (every day is a sub-problem), with the boundaries between the sub-problems being at hours 24, 48, and 72. Observe that the difference are not large for most of the intervals, resulting in the decomposed problem calculating a similar ESS dispatch. The total costs are only slightly affected, increasing by 0.4% as seen in Table I. If the number of sub-problems is increased to 16, the deviation in ESS dispatch becomes more significant, as shown in Fig. 3(b). In this case, each sub-problem only has an optimization horizon of 6 hours and the exchange of boundary variables is not sufficient to cope with the loss of information. Nevertheless, secure operation is achieved and the cost increase is not large (under 3%).

B. Computational Costs

The Newton step for a horizon of 96 time steps yields a KKT-system with a dimension of almost $10^6 \times 10^6$, and takes an average computing time of over 2 minutes, while this takes only 2 seconds for 6 time steps in the case of 16 sub-problems (Table I). Hence, there is a significant improvement in computation time when the problem is decomposed. Additionally, the number of iterations tends to decrease with a larger number of sub-problems. Note that the slowest sub-problem defines the overall computation time, which can be reduced from over 2 hours to a few minutes in the case of 8 sub-problems, as shown in Table I, while yielding a cost increase of less than 1%. When a Newton step is calculated within 2-3 seconds for more than 8 sub-problems, the total computation time cannot be reduced further, because communication overhead now plays an important role. Hence, based on the results shown in Table I, the optimal number of sub-problems would be 8, which can considerably reduce the run-time of the model, while not significantly increasing the operating cost with respect to the centralized approach (1 sub-problem).

In Fig. 4, the storage energy gap obtained in (49) is illustrated for all 3 boundaries when decomposing the problem into 4 sub-problems. Observe the evolution of the proposed approach through the three stages described in Section III-B. Thus, after the first stage, a gap of over 50% of the installed capacity can be observed, but due to the exchange of boundary variables it is reduced to below 3% after 15 iterations. After the boundaries are fixed in Stage III, it converges to zero.

The relative feasibility condition in (36) for all 4 sub-problems is depicted in Fig. 5. Note that this value quickly decreases to below 0.1 when the second stage starts, but the feasibility does not continuously decreases. This is due
A novel methodology has been presented to decouple large-scale SCOPF problems with inter-temporal ESS constraints into sub-problems to compute them efficiently in parallel using an SGCM system. A simple to implement three stage variable exchange process was proposed and described in detail. The approach was tested on a realistic model of the German transmission grid for 96 time steps. The results show that by implementing the SCOPF model in a distributed computing platform, the run-time can be considerably reduced, which is needed for time-dependent SCOPF studies of large-scale networks and online applications. A trade-off between the decreased computation time and increased system costs must be taken into account; for instance, in the proposed case study, the computation time can be reduced by a factor of 13 with a cost increase of 0.9% by choosing an adequate number of sub-problems.

V. CONCLUSIONS

A novel methodology has been presented to decouple large-scale SCOPF problems with inter-temporal ESS constraints into sub-problems to compute them efficiently in parallel using an SGCM system. A simple to implement three stage variable exchange process was proposed and described in detail. The approach was tested on a realistic model of the German transmission grid for 96 time steps. The results show that by implementing the SCOPF model in a distributed computing platform, the run-time can be considerably reduced, which is needed for time-dependent SCOPF studies of large-scale networks and online applications. A trade-off between the decreased computation time and increased system costs must be taken into account; for instance, in the proposed case study, the computation time can be reduced by a factor of 13 with a cost increase of 0.9% by choosing an adequate number of sub-problems.

ACKNOWLEDGMENT

The authors would like to thank Dr. Hongbing Fan from Wilfrid Laurier University, Waterloo, ON, Canada, for help and support with the implementation of the model on the SGCM system.

REFERENCES