IET Generation, Transmission & Distribution

Research Article

Time decomposition strategy for security-constrained economic dispatch

ISSN 1751-8687
Received on 27th October 2018
Revised 13th August 2019
Accepted on 18th September 2019
E-First on 25th October 2019
doi: 10.1049/iet-gtd.2018.6807
www.ietdl.org

Farnaz Safdarian[1], Amin Kargarian[1] ✉

[1]Department of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA 70803, USA

✉ *E-mail: kargarian@lsu.edu*

Abstract: A horizontal time decomposition strategy to reduce the computation time of security-constrained economic dispatch (SCED) is presented in this study. The proposed decomposition strategy is fundamentally novel and is developed in this paper for the first time. The considered scheduling horizon is decomposed into multiple smaller sub-horizons. The concept of overlapping time intervals is introduced to model ramp constraints for the transition from one sub-horizon to another sub-horizon. A sub-horizon includes several internal intervals and one or two overlapping time intervals that interconnect consecutive sub-horizons. A local SCED is formulated for each sub-horizon with respect to internal and overlapping intervals' variables/constraints. The overlapping intervals allow modelling intertemporal constraints between the consecutive sub-horizons in a distributed fashion. To coordinate the subproblems and find the optimal solution for the whole operation horizon distributedly, accelerated auxiliary problem principle is developed. Furthermore, the authors present an initialisation strategy to enhance the convergence performance of the coordination strategy. The proposed algorithm is applied to three large systems, and promising results are obtained.

Nomenclature

$i,\ j$	index for buses
t	index for time intervals
u	index for units
c	index for contingencies
k	index for iterations
ij	index for lines
N	index for sub-horizons
t_o	index for overlapping time intervals
n_{to}	number of overlapping time intervals between a sub-horizon and its neighbouring sub-horizons
N_-	sub-horizon (subproblem) before sub-horizon N
N_+	sub-horizon (subproblem) after sub-horizon N
n	number of time intervals in a sub-horizon
UR_u	ramping up of unit u
DR_u	ramping down of unit u
$p^{*(k-1)N_-}_{ut_oN_- \to N}$	power generated by unit u in overlapping time interval t_o between sub-horizons N_- and N determined at iteration $k-1$ by subproblem N_- and sent to subproblem N
$p^{*(k-1)N_+}_{ut_oN \to N_+}$	power generated by unit u in overlapping time interval t_o between sub-horizons N and N_+ determined at iteration $k-1$ by subproblem N_+ and sent to subproblem N
$f(\cdot)$	generation cost function
†	transpose operator
λ	vector of Lagrange multipliers
$\beta,\ \gamma$	tuning parameters
ρ	penalty factor
ϕ^*	predicted parameters in the accelerated APP
p_{utN}	generation of unit u at time t of sub-horizon N
p^c_{utN}	generation of unit u after contingency c at time t of sub-horizon N
δ_{tN}	voltage angle of bus i at time t of sub-horizon N
δ^c_{tN}	voltage angle of bus i after contingency c at time t of sub-horizon N
$p^{N_-}_{ut_oN_- \to N}$	decision variable in subproblem N_- for power generated by unit u in overlapping time interval t_o between sub-horizons N_- and N
$p^{N}_{ut_oN_- \to N}$	decision variable in subproblem N for power generated by unit u in overlapping time interval t_o between sub-horizons N_- and N

1 Introduction

Security-constrained economic dispatch (SCED) with generation ramp limits is a decision-making problem that is frequently solved for analysis of power systems. The considered scheduling time horizon may vary from days to weeks [1]. The size of multi-interval decision-making problems, such as ramp-constrained SCED, depends on the size of the considered system and the length of scheduling time intervals. The larger/more the system/number of time intervals is, the larger the size of the optimisation problem will be. Although ramp-constrained SCED is a tractable optimisation problem, its computational burden and solution time increase with growing the size of the problem [1]. Centralised methods and standard solvers, such as CPLEX, may face difficulties to handle such large optimisation problems within an acceptable time limit. Reducing the solution time of SCED is highly desirable as this problem is at the heart of many power system analyses, and decision-makers prefer to obtain solutions to their problems as fast as possible.

Approximations, relaxations, and decomposition techniques have been presented in the literature to alleviate the computational burden and find high-quality solutions within an acceptable range of time. Distributed optimisation algorithms are presented as promising techniques to decompose large problems into several subproblems and distribute computational burden on several computing machines [2]. Kargarian *et al*. [2] reviewed distributed and decentralised algorithms to solve the optimal power flow problem in electric power systems. Palomar and Chiang [3] provide a tutorial for decomposition methods for network utility maximisation. Molzahn *et al*. [4] survey the literature of distributed algorithms with applications to optimisation and control of power systems. In [5], the authors have overviewed distributed approaches, all based on consensus + innovations, for three common energy management functions: state estimation, economic dispatch (ED), and optimal power flow. Kargarian *et al*. [6] present a distributed unit commitment algorithm to accelerate the generation scheduling for large-scale power systems. Kargarian *et* *al*. [7] present a decentralised solution algorithm for network-constrained unit commitment in multiregional power systems. Liu and Chu [8] study iterative distributed algorithms to assess transfer capability for management of multi-area power systems. The authors of [9] have described a distributed implementation of optimal power flow on a network of workstations. An approach to paralleling optimal power flow is described in [10] that is applicable to large interconnected power systems. Authors of [11] and [12] present a coordination method for distributed ED problem. Distributed algorithms were implemented in [13] to solve large-scale unit commitment problems.

Many papers have tried to reduce SCED solution time to enhance decision-making processes for power system analysis problems. However, most techniques presented in the literature to solve ED in a distributed manner implement a geographical decomposition to divide the power system into several smaller subsystems (note that most of the existing geographical-based distributed algorithms in the literature aim at preserving information privacy of autonomous entities). Distributed algorithms, such as alternating direction method of multipliers [14], optimality condition decomposition [15], auxiliary problem principle [16], and analytical target cascading [17] are applied to coordinate the geographical subproblems and reach a good-enough solution from the perspective of the whole power system. Guo *et al*. [18] present a geographically distributed ED, where every generator/load is modelled as an agent, and the related agents of an area form a cluster. In [19], the authors have described a transmission + distribution ED and proposed a decentralised method to solve this problem using multi-parametric quadratic programming. Lagrangian relaxation and augmented Lagrangian relaxation are used in [20] to solve ED in a distributed fashion. Xu *et al*. [21] present a distributed ED with second-order convergence, which is based on a parallel primal-dual interior-point algorithm with a matrix-splitting technique. Chen and Zhao [22] proposed a consensus-based distributed ED algorithm and analyse the influence of time delays. An attack-robust distributed ED strategy was developed in [23] assuming that every distributed generator can monitor the behaviour of its neighbours.

However, the dimensionality and complexity of the ramp-constrained ED not only depend on the size of the system but also depend on the number of scheduling time intervals. Although decomposing the system geographically potentially reduces the size and computation time of the ED problem, it does not deal with intertemporal constraints. The intertemporal constraints, which originate from limits of ramping capabilities of generating units, interconnect decisions made in a time interval to decisions made in other intervals and increase the complexity of the decision-making process in the ramp-constrained ED. Another factor that increases the size and computational complexity of the ED problem is $N-1$ security criteria. The ED problem with $N-1$ security criteria is called SCED [24]. A set of credible contingencies is considered for each interval of the SCED, and a new set of variables and constraints are added to each time interval. Considering contingencies and interdependencies between scheduling time intervals (i.e. intertemporal constraints) increases the computation time of the SCED problem. In such a situation, decomposing SCED over the considered time horizon is potentially a promising strategy to reduce the computation time.

In this paper, we propose a distributed optimisation algorithm to reduce the computation time of SCED. A horizontal time decomposition strategy is developed to divide the ramp-constrained SCED problem into several smaller optimisation subproblems, each containing fewer intertemporal constraints and variables than the original centralised SCED. Unlike the geographical decomposition, the considered scheduling horizon is decomposed into several sub-horizons (we call this strategy as a horizontal time decomposition since it decomposes the optimisation over the time horizon with respect to intertemporal constraints). The concept of *overlapping time intervals* is presented to model intertemporal constraints, which are ramping constraints of generating units, for the transition from one sub-horizon to its neighbouring sub-horizons. The overlapping time intervals are mathematically modelled as a set of variables and constraints. An SCED subproblem is formulated for each sub-horizon with respect to variables and constraints of internal intervals and the overlapping intervals of that sub-horizon. Variables and constraints of an overlapping time interval can be controlled by two consecutive sub-horizons (an overlapping interval is at the end of sub-horizon N_{-} and at the beginning of sub-horizons N). To coordinate solutions of the SCED subproblems and ensure the feasibility and optimality of results from the perspective of the power system and its components, accelerated auxiliary problem principle (A-APP) is developed to solve subproblems distributedly. Furthermore, we present an initialisation technique to enhance the convergence performance of the distributed optimisation algorithm. The proposed algorithm is scalable and reduces the computation time of the SCED problem. The proposed horizontal decomposition and coordination strategies are applied to solve a week-ahead SCED problem on IEEE 118-bus system and a 472-bus system, and a day-ahead SCED on a 4720-bus system.

The main contributions of the paper, that make it different from the existing literature, are summarised as follows:

- A *temporal decomposition* strategy is proposed to decompose SCED into multiple time-interdependent subproblems. This strategy can be applied to other multi-stage problems. To the best of our knowledge, such a temporal decomposition strategy is its first kind developed for SCED.
- Based on the concept of Nesterov momentum approach for gradient descent methods, an *accelerated* APP is presented to coordinate SCED subproblem distributedly.
- An initialisation strategy is presented to enhance the performance of the proposed distributed SCED algorithm.

The proposed time decomposition and geographical decomposition are complements of each other. Geographical decomposition can be used to divide a system into several smaller subsystems. If the considered problem has multiple time intervals, time decomposition can be used to further decompose each subsystem into several sub-horizons to reduce the computational cost of each subsystem.

The rest of this paper is organised as follows. In Section 2, the considered SCED problem is formulated. In Section 3, a novel time decomposition strategy is proposed to decompose SCED over the time horizon. In Section 4, A-APP algorithm and an initialisation strategy are presented to solve SCED subproblems in parallel. Simulation results are illustrated in Section 5 and concluding remarks are provided in Section 6.

2 Considered SCED problem

The considered centralised SCED is formulated below. The objective function is to minimise generation costs subject to power flow, network, and equipment constraints under normal condition ((1b)–(1g)), and $N-1$ security criteria ((1h)–(1l)) [25].

$$\min_{(p_{ut},\, p_{ut}^{c},\, \delta_t)} \underbrace{\sum_t \sum_u a_u \cdot p_{ut}^2 + b_u \cdot p_{ut} + C_u}_{f(p_{ut})} \tag{1a}$$

$$\text{s.t.} \quad h(p,\delta): \begin{cases} p_{uit} - p_{dit} = \sum_j \dfrac{\delta_{it} - \delta_{jt}}{X_{ij}} & \forall i, \forall t \\ \delta_{\text{ref},t} = 0 & \forall t \end{cases} \tag{1b,c}$$

$$g(p,\delta): \begin{cases} \underline{P_{ut}} \le p_{ut} \le \overline{P_{ut}} & \forall\, u, \forall t \\ p_{ut} - p_{u(t-1)} \le \mathrm{UR}_u & \forall\, u, \forall t \\ p_{u(t-1)} - p_{ut} \le \mathrm{DR}_u & \forall\, u, \forall t \\ \underline{P_{ij}} \le \dfrac{\delta_{it} - \delta_{jt}}{X_{ij}} \le \overline{P_{ij}} & \forall ij, \forall t \end{cases} \tag{1d-g}$$

$$h^c(p,\delta):\begin{cases} p^c_{uit} - p_{dit} = \sum_j \dfrac{\delta^c_{it} - \delta^c_{jt}}{X_{ij}} & \forall i, \forall t, \forall c \\ \delta^c_{\text{ref},t} = 0 & \forall t, \forall c \end{cases} \quad \text{(1h-i)}$$

$$g^c(p,\delta):\begin{cases} \underline{P_{ut}} \le p^c_{ut} \le \overline{P_{ut}} & \forall u, \forall t, \forall c \\ \underline{P_{ij}} \le \dfrac{\delta^c_{it} - \delta^c_{jt}}{X_{ij}} \le \overline{P_{ij}} & \forall ij, \forall t, \forall c \\ |p_{ut} - p^c_{ut}| \le \Delta & \forall u, \forall t, \forall c \end{cases} \quad \text{(1j-l)}$$

We have used a DC power flow-based SCED formulation. However, a convex relaxation of the AC power flow model can be integrated to formulate a more complex and computationally expensive SCED formulation [26]. Although we have ignored uncertainties in the considered SCED formulation, chance constraints can be formulated to account for generation and demand uncertainties. In that case, line flow and power reserve inequality constraints should be modelled as chance constraints [27].

3 Proposed temporal decomposition strategy

Consider a scheduling horizon consisting of $n \times N$ intervals as shown in Fig. 1*a*. The horizon can be decomposed into N equal sub-horizons, each consisting of n intervals as depicted in Fig. 1*b*. One strategy is to formulate an SCED subproblem for each sub-horizon N regardless of other sub-horizons $N_{\mp}$ and then independently solve the subproblems in parallel. Another strategy is to formulate an SCED for each sub-horizon N by fixing the solution of interval n of SCED N_{-} as the initial condition of SCED N and then solve SCED subproblems 1 to N sequentially. While the first strategy may violate system intertemporal constraints between the ending interval of sub-horizon N_{-} and the beginning interval of sub-horizon N, the second strategy may provide feasible but suboptimal results for the whole scheduling horizon $n \times N$. Consider the SCED subproblems of two consecutive sub-horizons N_{-} and N (from now on, for the sake of brevity, we write SCED of (1) in a compact form).

$$\min_{(p_{N\_}, p^c_{N\_}, \delta_{N\_}, \delta^c_{N\_})} \sum_{t=1}^{n} \sum_u f(p_{utN\_}) \quad \text{(2a)}$$

$$\text{s.t.} \quad h_{N\_}(p_{utN\_},\ p^c_{utN\_},\ \delta_{itN\_},\ \delta^c_{itN\_}) = 0 \quad \forall t, u, i, c \quad \text{(2b)}$$

$$g_{N\_}(p_{utN\_}, p^c_{utN\_}, \delta_{itN\_}, \delta^c_{itN\_}) \le 0 \quad \forall t, u, i, c \quad \text{(2c)}$$

where equality constraint (2b) is the compact form of (1b), (1c), (1h) and (1i), and inequality constraint (2c) is the compact form of (1d)–(1g) and (1j)–(1l) for SCED corresponding to sub-horizons N_{-}.

$$\min_{(p_N, p^c_N, \delta_N, \delta^c_N)} \sum_{t=1}^{n} \sum_u f(p_{utN}) \quad \text{(3a)}$$

$$\text{s.t.} \quad h_N(p_{utN}, p^c_{utN}, \delta_{itN}, \delta^c_{itN}) = 0 \quad \forall t, u, i, c \quad \text{(3b)}$$

$$g_N(p_{utN}, p^c_{utN}, \delta_{itN}, \delta^c_{itN}) \le 0 \quad \forall t, u, i, c \quad \text{(3c)}$$

Equality constraint (3b) is the compact form of (1b), (1c), (1h) and (1i), and inequality constraint (3c) is the compact form of (1d)–(1g) and (1j)–(1l) for SCED corresponding to sub-horizons N.

As shown in Fig. 2, the constraints of sub-horizon N_{-} are coupled with the constraints of sub-horizon N through the generators' ramping constraints for the transition from interval n of sub-horizon N_{-} to interval one of sub-horizon N. The red (blue) line indicates that if the unit u is generating $p_{unN\_}$ at the end of sub-horizon N_{-}, it cannot generate more (less) than $p_{unN\_} + UR_u$ ($p_{unN\_} - DR_u$) at the beginning of sub-horizon N. The coupling constraints between sub-horizons N_{-} and N are as shown in the equations presented in Fig. 2.

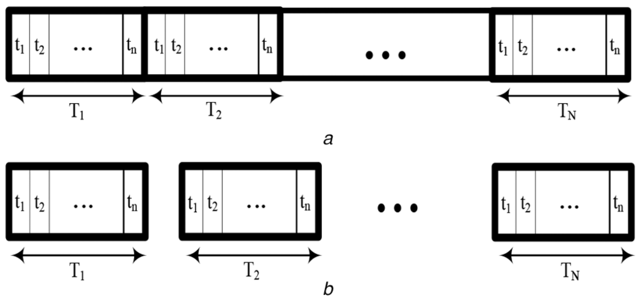

Fig. 1 *Time decomposition*
(a) Whole $n \times N$ intervals, and ***(b)*** decomposed N sub-horizons each with n intervals

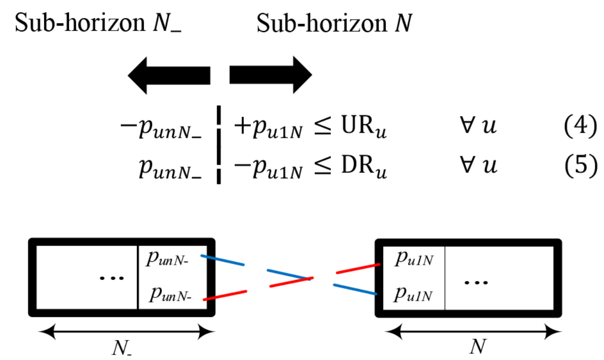

Fig. 2 *Coupling constraint (ramping limit) for transition from sub-horizon N_{-} to sub-horizon N*

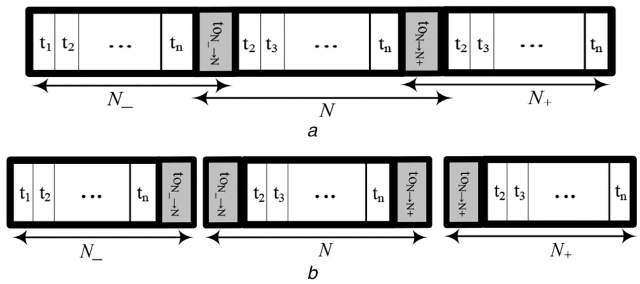

Fig. 3 *Time decomposition for three consecutive sub-horizons with overlapping intervals*
(a) The whole time horizon, ***(b)*** Sub-horizons decomposition over time horizon

To handle these two sets of coupling intertemporal constraints, we present the concept of overlapping (or coupling) time intervals between the consecutive sub-horizons as illustrated in Fig. 3*a*. We duplicate the first interval of each sub-horizon N and consider that as interval $n+1$ of sub-horizon N_{-}. All variables and constraints of the overlapping time intervals appear in both optimisation subproblems corresponding to sub-horizons N_{-} and N. Coordinating the time intervals n and $n+1$ of sub-horizon N_{-} and the time intervals one and two of sub-horizon N through intertemporal constraints inside each sub-horizon leads to coordination between the consecutive sub-horizons. We separate the sub-horizons and duplicate the overlapping time intervals in each sub-horizon as shown in Fig. 3*b*.

The optimisation subproblem N_{-} (i.e. (2a)–(2c)) with the overlapping time intervals is now formulated as follows:

$$\min_{(p_{N\_}, p^c_{N\_}, \delta_{N\_}, \delta^c_{N\_}, p_{t_o N\_ \to N})} \sum_{t=1}^{n+n_{to}} \sum_u f\left(p_{utN\_}, p^{N\_}_{ut_o N\_ \to N}\right) \quad \text{(6a)}$$

$$\text{s.t.} \quad \text{(2b), (2c)}$$

$$N_{-} \rightarrow N \begin{cases} h_{N\_}^{to}\left(p_{utN\_},\ p_{utN\_}^{c}, \delta_{itN\_}, \delta_{itN\_}^{c}, p_{ut_oN\_ \rightarrow N}^{N\_}\right) = 0 \\ \forall u, i, c \\ g_{N\_}^{to}\left(p_{utN\_},\ p_{utN\_}^{c}, \delta_{itN\_}, \delta_{itN\_}^{c}, p_{ut_oN\_ \rightarrow N}^{N\_}\right) \le 0 \\ \forall u, i, c \end{cases} \quad (6b,c)$$

where (2b) and (2c) are constraints corresponding to internal intervals of sub-horizon $N\_$, and (6b) and (6c) are constraints corresponding to the overlapping time intervals for the transition from sub-horizon $N\_$ to sub-horizon N. The optimisation subproblem N (i.e. (3a)–(3c)), which is a middle subproblem with two overlapping time intervals at the beginning and ending time intervals, is formulated as

$$\min_{\left(p_{t_oN\_ \rightarrow N}, p_N, p_N^c, \delta_N, \delta_N^c, p_{t_oN \rightarrow N_+}\right)} \sum_{t=1}^{n+n_{to}} \sum_u f\begin{pmatrix} p_{ut_oN\_ \rightarrow N}^{N}, p_{utN}, \\ p_{ut_oN\_ \rightarrow N}^{N} \end{pmatrix} \quad (7a)$$

$$\text{s.t.} \quad (3b), (3c)$$

$$N_{-} \rightarrow N \begin{cases} h_N^{to}\left(p_{ut_oN\_ \rightarrow N}^{N}, p_{utN},\ p_{utN}^{c}, \delta_{itN}, \delta_{itN}^{c}\right) = 0 \\ \forall u, i, c \\ g_N^{to}\left(p_{ut_oN\_ \rightarrow N}^{N}, p_{utN},\ p_{utN}^{c}, \delta_{itN}, \delta_{itN}^{c}\right) \le 0 \\ \forall u, i, c \end{cases} \quad (7b,c)$$

$$N \rightarrow N_{+} \begin{cases} h_N^{to}\left(p_{utN},\ p_{utN}^{c}, \delta_{itN}, \delta_{itN}^{c}, p_{ut_oN \rightarrow N_+}^{N}\right) = 0 \\ \forall u, i, c \\ g_N^{to}\left(p_{utN},\ p_{utN}^{c}, \delta_{itN}, \delta_{itN}^{c}, p_{ut_oN \rightarrow N_+}^{N}\right) \le 0 \\ \forall u, i, c \end{cases} \quad (7d,e)$$

where (3b) and (3c) are constraints corresponding to internal intervals of sub-horizon N. Since sub-horizon N has overlapping intervals with sub-horizons $N\_$ and N_+, its optimisation subproblem includes two sets of overlapping constraints, one with $N\_$ and another one with N_+. Expressions (7b) and (7c) are constraints corresponding to the overlapping time intervals for the transition from sub-horizon $N\_$ to sub-horizon N, and (7d) and (7e) are constraints corresponding to the overlapping time intervals for the transition from sub-horizon N to sub-horizon N_+. The optimisation subproblems (6) and (7) are still non-separable as they include shared variables $p_{ut_oN\_ \rightarrow N}$. In addition, subproblems N and N_+ are non-separable because of shared variables $p_{ut_oN \rightarrow N_+}$. The shared variables are power produced by generating units in the overlapping time intervals. We duplicate $p_{ut_oN\_ \rightarrow N}$ to create two sets of new variables with one belonging to subproblem N_- ($p_{ut_oN\_ \rightarrow N}^{N\_}$) and another one belonging to subproblem N ($p_{ut_oN\_ \rightarrow N}^{N}$). Similarly, we duplicate $p_{ut_oN \rightarrow N_+}$ and assign one set to subproblem N ($p_{ut_oN \rightarrow N_+}^{N}$) and the other set to subproblem N_+ ($p_{ut_oN \rightarrow N_+}^{N_+}$). These auxiliary variables make SCEDs of the sub-horizons separable. That is, each SCED subproblem can be solved separately from other subproblems. To ensure the feasibility of the results from the perspective of the whole scheduling horizon, we convert each pair of the shared variables to a set of consistency constraint as

$$CC_{N\_ \rightarrow N}\colon\ p_{ut_oN\_ \rightarrow N}^{N\_} - p_{ut_oN\_ \rightarrow N}^{N} = 0 \quad \forall u \quad (8)$$

$$CC_{N \rightarrow N_+}\colon p_{ut_oN \rightarrow N_+}^{N_+} - p_{ut_oN \rightarrow N_+}^{N_+} = 0 \quad \forall u \quad (9)$$

$CC_{N\_ \rightarrow N}$ is enforced in subproblems N_- and N, and $CC_{N \rightarrow N_+}$ is enforced in subproblems N and N_+. The SCED subproblem of sub-horizon N_- is

$$\min\ (6a)$$

$$\text{s.t.} \quad (2b),\ (2c),\ (6b),\ (6c),\ (8)$$

and the SCED subproblem of sub-horizon N is

$$\min\ (7a)$$

$$\text{s.t.} \quad (3b),\ (3c),\ (7b)-(7e),\ (8),\ (9)$$

While $p_{ut_oN\_ \rightarrow N}^{N\_}$ is a decision variable in subproblem N_-, it is treated as a constant in subproblem N. Similarly, $p_{ut_oN\_ \rightarrow N}^{N_+}$ is a decision variable in subproblem N_+ and a constant in subproblem N. Consistency constraints (8) and (9) are hard constraints that make the order of solving subproblems of importance. If sub-horizon N_- solves its SCED first, $CC_{N\_ \rightarrow N}$ forces sub-horizon N to set its decision variables $p_{ut_oN\_ \rightarrow N}^{N}$ equal to $p_{ut_oN\_ \rightarrow N}^{N\_}$. Since the sub-horizons are connected like a string, if one starts from the beginning of the string to solve the decomposed SCED problem, the solution of hour one of each sub-horizon is dictated by the solution of its previous sub-horizon. In the existing literature and current practice, the initial condition of every sub-horizon (which is usually one day in ramp-constrained SCED) is set by its previous sub-horizon. However, this may lead to a suboptimal solution for the whole scheduling horizon.

4 Proposed coordination algorithm

To avoid obtaining suboptimal results, we present an iterative coordination algorithm that allows a parallel solution of the SCED subproblems. The consistency constraints are relaxed in the objective function of each sub-horizon using the concept of augmented Lagrangian relaxation. We utilise the auxiliary problem principle [16], which is a parallel coordination strategy, to coordinate the sub-horizons. We further propose an accelerated APP (A-APP) and an initialisation strategy to enhance the performance of the distributed coordination algorithm.

4.1 Normal APP

APP is an iterative method that aims at finding the optimal solution of several coupled optimisation problems in a distributed manner [16]. APP is a suitable method for parallel computing and reduces the idle time of subproblems as compared to sequential distributed approaches. This method uses the augmented Lagrangian relaxation to penalise the consistency constraints (i.e. hard constraints) with a set of penalty terms in the objective function. A sequence of auxiliary problems are solved to make the non-separable terms of the augmented Lagrangian (i.e. the quadratic term) separable and solve subproblems in parallel (we refer to [2] for more details).

Consider the SCED subproblems N_- and N. We denote the iteration index of APP as k and penalise the violation of hard constraint (8) at each iteration k in the objective functions of subproblem N_-. The resultant SCED subproblem for sub-horizon N_- becomes as follows:

$$\min_{\left(p_{N\_}^{k}, p_{N\_}^{kc}, \delta_{N\_}, \delta_{N\_}^{c}, p_{t_oN\_ \rightarrow N}^{kN\_}\right)} \sum_{t=1}^{n+n_{to}} \sum_u f\left(p_{utN\_}^{k},\ p_{ut_oN\_ \rightarrow N}^{kN\_}\right)$$
$$+ \sum_u \begin{pmatrix} \frac{\rho}{2} p_{ut_oN\_ \rightarrow N}^{kN\_} - p_{ut_oN\_ \rightarrow N}^{*(k-1)N\_\,2} + \lambda_{N\_ \rightarrow N}^{k\dagger} p_{ut_oN\_ \rightarrow N}^{kN\_} \\ + \gamma p_{ut_oN\_ \rightarrow N}^{kN\_\dagger}\left(p_{ut_oN\_ \rightarrow N}^{*(k-1)N\_} - p_{ut_oN\_ \rightarrow N}^{*(k-1)N}\right) \end{pmatrix} \quad (10)$$

$$\text{s.t.}\ (2b),\ (2c),\ (6b),\ (6c)$$

where $p_{utN\_}^{k}$, $p_{utN\_}^{kc}$, and $p_{ut_oN\_ \rightarrow N}^{kN\_}$ are decision variables while $p_{ut_oN\_ \rightarrow N}^{*(k-1)N}$ and $p_{ut_oN\_ \rightarrow N}^{*(k-1)N\_}$ are known. Parameter $p_{ut_oN\_ \rightarrow N}^{*(k-1)N}$ is the value of generated power by each unit u (i.e. the shared variable) in the overlapping time interval determined by sub-horizon N at iteration $k-1$. Parameter $p_{ut_oN\_ \rightarrow N}^{*(k-1)N\_}$ denotes the value of generated power by each unit u in the overlapping time interval determined by sub-horizon N_- at iteration $k-1$. The SCED subproblem N is penalised by relaxing two sets of hard constraints (8) and (9) in its objective function as follows:

$$\begin{aligned}&\min_{(p^k_{t_oN_-\to N},\, p^k_N,\, p^{kc}_N,\, \delta_N,\, \delta^c_N,\, p^k_{t_oN\to N_+})} \sum_{t=1}^{n+n_{to}} \sum_u f\begin{pmatrix} p^{kN}_{ut_cN_-\to N}, \\ p^k_{utN},\;\; p^{kN}_{ut_oN\to N_+}\end{pmatrix} \\ &+ \sum_u \left(\begin{pmatrix} \frac{\rho}{2} p^{kN}_{ut_oN_-\to N} - p^{*(k-1)N}_{ut_oN_-\to N}{}^2 - \lambda^{k\dagger}_{N_-\to N} p^{kN}_{ut_oN_-\to N} \\ +\gamma p^{kN\dagger}_{ut_oN_-\to N}\left(p^{*(k-1)N}_{ut_oN_-\to N} - p^{*(k-1)N_-}_{ut_oN_-\to N}\right) \end{pmatrix} \right. \\ &\left. + \begin{pmatrix} \frac{\rho}{2} p^{kN}_{ut_oN\to N_+} - p^{*(k-1)N}_{ut_oN\to N_+}{}^2 + \lambda^{k\dagger}_{N\to N_+} p^{kN}_{ut_oN\to N_+} \\ +\gamma p^{kN\dagger}_{ut_oN\to N_+}\left(p^{*(k-1)N}_{ut_oN\to N_+} - p^{*(k-1)N_+}_{ut_oN\to N_+}\right) \end{pmatrix} \right) \\ &\text{s.t. (3b), (3c), (7b)} - \text{(7e)}\end{aligned} \tag{11}$$

While the first penalty term penalises any violations in the overlapping time interval between sub-horizons N_- and N, the second term penalises any violations in power produced by each unit u at the overlapping time interval between sub-horizons N and N_+. Since in each iteration k, each sub-horizon N uses values of the shared variables obtained by its neighbouring sub-horizons N_- and N_+ at iteration $k-1$ (i.e. $p^{*(k-1)N_-}_{ut_oN_-\to N}$ and $p^{*(k-1)N_+}_{ut_oN\to N_+}$), the SCED subproblems can be solved in parallel. Before starting each iteration k, the penalty multiplier λ is updated as follows:

$$\lambda^{k+1}_{N_-\to N} = \lambda^k_{N_-\to N} + \beta\left(p^{*kN_-}_{ut_oN_-\to N} - p^{*kN}_{ut_oN_-\to N}\right) \tag{12}$$

$$\lambda^{k+1}_{N\to N_+} = \lambda^k_{N\to N_+} + \beta\left(p^{*kN}_{ut_oN\to N_+} - p^{*kN_+}_{ut_oN\to N_+}\right) \tag{13}$$

where β should be a suitable positive step-size. If the step-size is selected too large, the convergence speed may go up; however, it increases the chance of losing optimality, oscillating around the optimal point, or even divergence. If the step-size is selected too small, although the solution is more accurate and the chance of divergence reduces, the convergence speed degrades [28]. A user may solve the problem several times with different step-sizes to gain knowledge on the suitable values/ranges for the step-size. The value of the Lagrange multiplier λ in each iteration implies the cost to maintain the consistency constraints in the overlapping time intervals. The above formulation can be generalised for multiple sub-horizons. The pseudocode for implementation of distributed SCED is given in Algorithm 1 (see Fig. 4). Since the considered SCED problem is convex, the APP is proven to converge to the global optimal solution. We refer to [16] for discussions on the convergence rate/proof of APP.

4.2 Accelerated auxiliary problem principle

The normal APP uses a simple fixed step size gradient-based method to update multipliers with respect to the difference between coupling variables $\left(p^{kN_-}_{ut_oN_-\to N} - p^{kN}_{ut_oN_-\to N} \;\&\; p^{kN}_{ut_oN\to N_+} - p^{kN_+}_{ut_oN\to N_+}\right)$, which is the gradient. However, the convergence performance might degrade when more iterations are carried out, and the solution is getting close to the optimal point or when the optimal point is in a shallow area (or a ravine). Near the optimal point, the terms $p^{kN_-}_{ut_oN_-\to N} - p^{kN}_{ut_oN_-\to N}$ and $p^{kN}_{ut_oN\to N_+} - p^{kN_+}_{ut_oN\to N_+}$ (or gradients) might become very small that leads to an updated multiplier in iteration $k+1$ that is almost the same as that in iteration k, i.e. $\lambda^{k+1} \approx \lambda^k$. This slows the convergence speed.

The local objective functions of the SCED subproblems are strongly convex. Taking advantages of this feature, we present an accelerated APP based on the technique that was first proposed by Nesterov to accelerate gradient descent methods and later was used to accelerate alternating direction method of multipliers [29, 30]. The suggested accelerated APP utilises a prediction type acceleration step. The momentum of the algorithm is used to prevent the algorithm from deceleration while more iterations are

Decompose the considered horizon into T_N equal sub-horizons
Initialize $p^{*0N_-}_{ut_oN_-\to N}$, $p^{*0N}_{ut_oN_-\to N}$, $p^{*0N}_{ut_oN\to N_+}$, $p^{*0N_+}_{ut_oN\to N_+}$, $\lambda, \beta, \gamma, \rho$ and set $k=0$
while $\left|p^{*kN_-}_{ut_oN_-\to N} - p^{*kN}_{ut_oN_-\to N}\right| > \varepsilon$ & $\left|p^{*kN}_{ut_oN\to N_+} - p^{*kN_+}_{ut_oN\to N_+}\right| > \varepsilon$, $k = k+1$ **do**
    Solve SCED subproblems in parallel and determine the optimal values of $p^{kN_-}_{ut_oN_-\to N}$, $p^{kN}_{ut_oN_-\to N}$, $p^{kN}_{ut_oN\to N_+}$, and $p^{kN_+}_{ut_oN\to N_+}$
    Exchange $p^{*kN_-}_{ut_oN_-\to N}$, $p^{*kN}_{ut_oN_-\to N}$, $p^{*kN}_{ut_oN\to N_+}$, and $p^{*kN_+}_{ut_oN\to N_+}$ between the neighboring sub-horizons
    Update λ^k by (12) and (13)
end while

Fig. 4 *Algorithm 1 Pseudocode for coordinating SCED subproblems with normal APP*

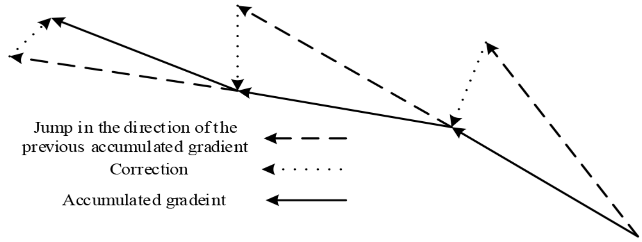

Fig. 5 *Nesterov update*

carried out. After each iteration k, the cumulated gradient of the previous iterations (i.e. momentum) is calculated as the predicted direction of the algorithm in the next iteration, and a big jump is made in that direction. Then, the gradient is measured, and a correction is made to avoid going too fast. This procedure is shown in Fig. 5. The dashed line shows the jump in the direction of the previously accumulated gradient, the dotted line indicates the correction based on measuring the gradient in the new point, and the solid line is the accumulated gradient (predicted-correction direction), used in the accelerated APP algorithm [31].

To implement the accelerated APP for SCED, the penalty term in the objective function (10) at iteration k is modified as follows:

$$\sum_u \begin{pmatrix} \frac{\rho}{2} p^{kN_-}_{ut_oN_-\to N} - \phi^{*(k-1)N_-}_{ut_oN_-\to N}{}^2 + \hat{\lambda}^{k\dagger}_{N_-\to N} p^{kN_-}_{ut_oN_-\to N} \\ +\gamma p^{kN_-\dagger}_{ut_oN_-\to N}\left(\phi^{*(k-1)N_-}_{ut_oN_-\to N} - \phi^{*(k-1)N}_{ut_oN_-\to N}\right) \end{pmatrix} \tag{14}$$

in which $p^{*(k-1)N_-}_{ut_oN_-\to N}$, $p^{*(k-1)N}_{ut_oN_-\to N}$, and $\lambda_{N_-\to N}$ are replaced by $\phi^{*(k-1)N_-}_{ut_oN_-\to N}$, $\phi^{*(k-1)N}_{ut_oN_-\to N}$, and $\hat{\lambda}_{N_-\to N}$. The two penalty terms in objective function (11) at iteration k are modified by replacing $p^{*(k-1)N_-}_{ut_oN_-\to N}$, $p^{*(k-1)N}_{ut_oN_-\to N}$, $p^{*(k-1)N}_{ut_oN\to N_+}$, $p^{*(k-1)N_+}_{ut_oN\to N_+}$, $\lambda_{N_-\to N}$, and $\lambda_{N\to N_+}$ by $\phi^{*(k-1)N_-}_{ut_oN_-\to N}$, $\phi^{*(k-1)N}_{ut_oN_-\to N}$, $\phi^{*(k-1)N}_{ut_oN\to N_+}$, $\phi^{*(k-1)N_+}_{ut_oN\to N_+}$, $\hat{\lambda}_{N_-\to N}$, and $\hat{\lambda}_{N\to N_+}$, respectively.

$$\begin{aligned}&\sum_u \left(\begin{pmatrix} \frac{\rho}{2} p^{kN}_{ut_oN_-\to N} - \phi^{*(k-1)N}_{ut_oN_-\to N}{}^2 - \hat{\lambda}^{k\dagger}_{N_-\to N} p^{kN}_{ut_oN_-\to N} \\ +\gamma p^{kN\dagger}_{ut_oN_-\to N}\left(\phi^{*(k-1)N}_{ut_oN_-\to N} - \phi^{*(k-1)N_-}_{ut_oN_-\to N}\right) \end{pmatrix} \right. \\ &\left. + \begin{pmatrix} \frac{\rho}{2} p^{kN}_{ut_oN\to N_+} - \phi^{*(k-1)N}_{ut_oN\to N_+}{}^2 + \hat{\lambda}^{k\dagger}_{N\to N_+} p^{kN}_{ut_oN\to N_+} \\ +\gamma p^{kN\dagger}_{ut_oN\to N_+}\left(\phi^{*(k-1)N}_{ut_oN\to N_+} - \phi^{*(k-1)N_+}_{ut_oN\to N_+}\right) \end{pmatrix} \right)\end{aligned} \tag{15}$$

The new parameters marked by ($\hat{}$) are predictions of actual shared variables/penalty multipliers. The predicted values that are used at iteration $k+1$ are calculated based on the actual shared variables and penalty multipliers obtained at iterations k and $k-1$. ϕ^{*k+1} and $\hat{\lambda}^{k+1}$ are updated according to (17) and (18), respectively.

$$\alpha_{k+1} = \frac{1+\sqrt{1+4\alpha_k^2}}{2} \tag{16}$$

$$\phi^{*k+1} = p^{*k} + \frac{\alpha_k - 1}{\alpha_{k+1}}(p^{*k} - p^{*(k-1)}) \tag{17}$$

$$\hat{\lambda}^{k+1} = \lambda^k + \frac{\alpha_k - 1}{\alpha_{k+1}}(\lambda^k - \lambda^{k-1}) \tag{18}$$

The solution procedure and the way that we calculate the predicted parameters are shown in Algorithm 2 (see Fig. 6).

4.3 Initialisation

One of the main drawbacks of distributed and decentralised optimisation algorithms is their dependency on initial conditions. That is, the convergence performance (the number of iterations and the optimality gap) might be different if two different sets of initial conditions are used. If a set of good initial conditions is selected, the proposed APP-based distributed SCED potentially converges in a considerably fewer number of iterations as compared to a case in which good initial conditions are not available. A good initial condition is a system- and problem-dependent. This is ongoing research in power systems and operations research communities.

Since the main goal of the proposed time decomposition is to decrease the computation time, we need to choose a suitable starting point. For this purpose, we take advantage of power system characteristics and propose a technique to find a set of good initialise values for the shared variables at the overlapping time intervals. To initialise the problem, we ignore the overlapping time intervals and shared variables (i.e. the sub-horizons are independent) and solve the SCED subproblems in parallel. Intuitively, since the load does not drastically change by the transition from the last time interval of sub-horizon N_- to the first interval of sub-horizon N, ignoring ramp rates of the units (which are eliminated as the shared variables) does not impose a large error to the solution. Although the obtained results might not be feasible, they are close to the feasible solution, and the relative error might be small from the beginning. We use the results of this procedure to initialise the APP-based distributed SCED. The pseudocode for the distributed SCED with the initialisation technique is given in Algorithm 3 (see Fig. 7).

1: Decompose the considered horizon into T_N equal sub-horizons
2: Initialize $\phi^{*0N_-}_{ut_oN_-\to N}$, $\phi^{*0N}_{ut_oN_-\to N}$, $\phi^{*0N}_{ut_oN\to N_+}$, and $\phi^{*0N_+}_{ut_oN\to N_+}$, $\hat{\lambda}, \beta, \gamma, \rho$ and set $k = 0$
3: **while** $|\phi^{*kN_-}_{ut_oN_-\to N} - \phi^{*kN}_{ut_oN_-\to N}| > \varepsilon$ & $|\phi^{*kN}_{ut_oN\to N_+} - \phi^{*kN_+}_{ut_oN\to N_+}| > \varepsilon$, $k = k + 1$ **do**
4: Solve SCED subproblems in parallel and determine the optimal values of $p^{*kN_-}_{ut_oN_-\to N}$, $p^{*kN}_{ut_oN_-\to N}$, $p^{*kN}_{ut_oN\to N_+}$, and $p^{*kN_+}_{ut_oN\to N_+}$
5: Exchange $p^{*kN_-}_{ut_oN_-\to N}$, $p^{*kN}_{ut_oN_-\to N}$, $p^{*kN}_{ut_oN\to N_+}$, and $p^{*kN_+}_{ut_oN\to N_+}$ between the SCED subproblems
6: Update λ^k by (12) and (13)
7: Calculate α_{k+1} by (16)
8: Update ϕ^{*k+1} and $\hat{\lambda}^{k+1}$ by (17) and (18)
9: **end while**

Fig. 6 *Algorithm 2 Pseudocode for coordinating SCED subproblems with accelerated APP*

1: Decompose the considered horizon into T_N equal sub-horizons
2: Ignore the overlapping time intervals and shared variables
3: Solve the SCED subproblems in parallel
4: Use the obtained results to initialize $\phi^{*0N_-}_{ut_oN_-\to N}$, $\phi^{*0N}_{ut_oN_-\to N}$, $\phi^{*0N}_{ut_oN\to N_+}$, and $\phi^{*0N_+}_{ut_oN\to N_+}$
5: Set multipliers $\lambda^0, \beta, \gamma, \rho$ and set $k = 0$
6: Do the while loop of Algorithm I for the normal APP or Algorithm II for the accelerated APP

Fig. 7 *Algorithm 3 Pseudocode for coordinating SCED subproblems with APP + initialisation*

4.4 Discussion on convergence

The Nesterov momentum technique is proven to accelerate the convergence of gradient descent methods [32]. On the other hand, APP, that is a gradient descent-based method, is proven to converge for convex problems (the considered SCED problem in this paper is convex) [33]. Hence, intuitively, using the Nesterov accelerated technique instead of the ordinary gradient descent for updating Lagrange multipliers after each iteration of APP keeps the convergence proof of APP valid and enhances its convergence speed.

5 Case study

The proposed algorithm is applied to solve a week-ahead SCED problem for the IEEE 118-bus system and a 472-bus system and a day-ahead problem for a 4720-bus test system. System and equipment characteristics are given in [32]. All simulations are carried out on Matlab using YALMIP [34] as modelling software and the QP solver of ILOG CPLEX 12.4 on a 3.7 GHz personal computer with 16 GB of RAM. The solver default settings are used. Although the reported results are obtained by CPLEX, we have applied Gurobi and Mosek and achieved almost the same simulation times as those reported in this paper.

To evaluate the performance of the proposed distributed SCED algorithm, we use a convergence index rel that shows the relative difference between the operating costs determined by the distributed SCED (f^d) and the centralised SCED (f^*) (note that f^* is the benchmark cost)

$$\text{rel} = \frac{|f^* - f^d|}{f^*} \tag{19}$$

The closer the convergence measure gets to zero, a more precise solution is obtained. We have found that the choice of 1% maximum mismatch between the shared variables yields a high-quality solution with a negligible rel index for all studied cases. This maximum mismatch is considered as the stopping criteria. Note that unlike the geographical decomposition, the nodal power balance is always satisfied at each iteration of the time decomposition strategy.

5.1 IEEE 118-Bus system

Six cases and sensitivity analysis are studied. The operation horizon is divided into seven sub-horizons, each including 24 intervals. To have a fair comparison, we set $\beta,\ \rho,\ \gamma = 0.2$ and $\lambda = 1$ in all cases except for the sensitivity analysis in which we study the effect of multipliers on the performance of the algorithm.

Case 1: Four contingency scenarios are considered for each hour (note that one can consider a complete set of $N-1$ contingencies. As shown in further cases, the efficiency of the proposed distributed SCED as compared to the centralised SCED becomes better for larger problems). We use a flat start strategy to initialise the shared variables (power generated by units in overlapping intervals) as $p^{N}_{t_o\bar{N}_- \to N} = P_{\min}$ and $p^{N}_{t_oN_- \to N} = P_{\max}$. As shown in Table 1, the proposed distributed SCED converges after nine iterations within 11 s. The operation cost is \$11,099,286. Fig. 8 shows the rel values over the course of iterations and time. The convergence measure decreases as more iterations are carried out. The rel index is 0.0004 upon the algorithm convergence (one may run the algorithm for more iterations to get a smaller rel index; however, we stop the algorithm at this point since we further apply the proposed initialisation strategy to significantly reduce error to 1×10^{-8} as shown in Table 1).

We implement the accelerated distributed algorithm. The results are shown in Table 1, and the rel index is depicted in Fig. 8. Although the number of iterations and the required time to converge are the same as the normal distributed algorithm, 75% of improvement is observed in rel. The rel index obtained by the accelerated algorithm is less than that for the normal algorithm over the course of iterations. If a user wants to reach a specific rel

Table 1 Comparison of rel and simulation times obtained by distributed and centralised SCED approaches for Cases 1 and 2

Case no.	Algorithm	Iteration	rel	Time, s
	centralised	—	—	14
Case 1	APP	9	0.0004	11
	A-APP	9	0.0001	11
Case 2	APP + initialisation	2 + 1	1×10^{-8} (≈ 0)	5
	A-APP + initialisation	2 + 1	110^{-8} (≈ 0)	5

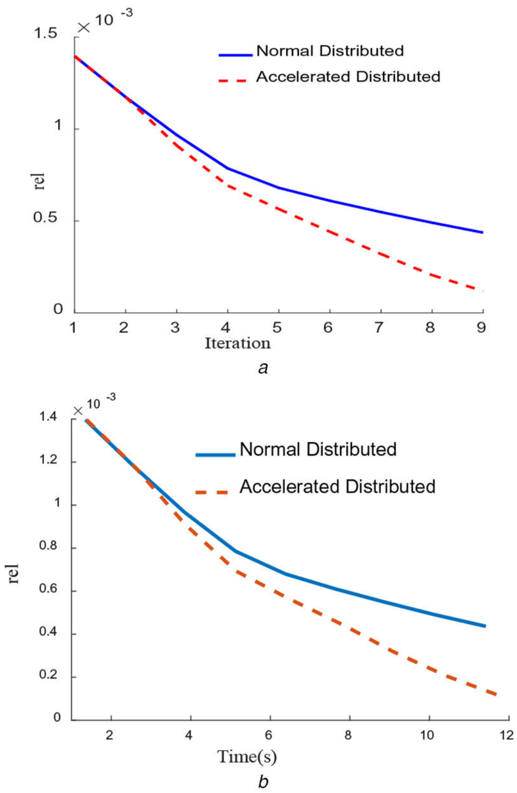

Fig. 8 *The convergence measure for case 1*
***(a)** Rel* vs. iterations, ***(b)** Rel* vs. time

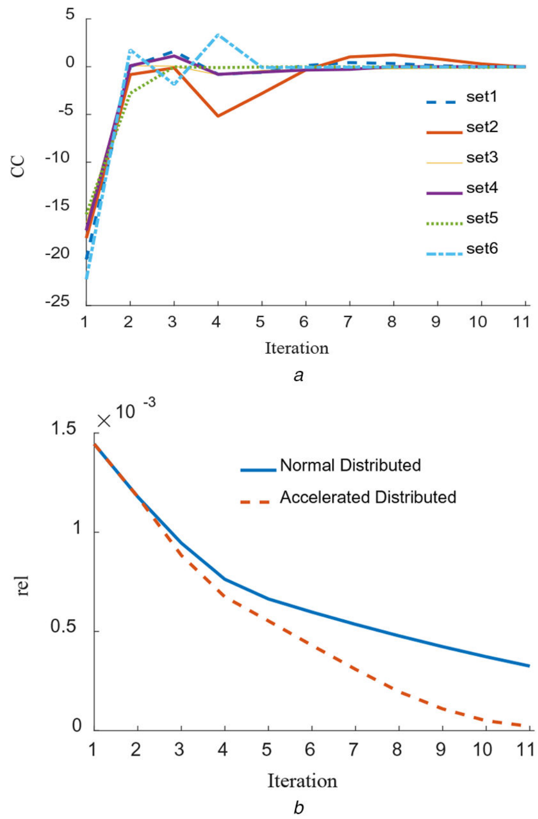

Fig. 9 *Two convergence measures for case 3*
(a) Consistency constraints corresponding to unit 25 vs. iterations, ***(b)** rel* vs. iterations

index (e.g. 0.0002), the normal APP takes more iterations and time to reach that target rel compared to the accelerated APP.

Case 2: The proposed initialisation strategy is implemented. All other values are the same as in Case 1. As shown in Table 1, the distributed SCED takes 5 s to converge after 2 + 1 iterations. Note, 2 + 1 means that the algorithm takes two iterations in addition to the one iteration needed for initialisation. We have counted the computation time of the initialisation step in the reported times. The number of iterations is six less than that for Case 1, and the computation time has been reduced by 55%. Also, the relative error has been considerably decreased to almost zero. The same trend is observed for the accelerated SCED. Note that in this case, both normal and accelerated algorithms provide almost the same results. This is because of the very high accuracy of the solution because of the initialisation step.

Case 3: This case is similar to Case 1 except for the permissible ramping of several units that are purposely reduced to show that the proposed algorithm handles problems of which shared variables in the neighbouring sub-horizons have a considerable difference at initialisation (this makes the SCED problem more computationally complex). Several intertemporal constraints are heavily binding. This condition affects Lagrange multipliers and can be interpreted as a situation in which lines are congested in a geographical decomposition. In such a case, the values of Lagrange multipliers are larger than those for Case 1, and thus the sensitivity of the solution to the variations of shared variables is higher. Fig. 9*a* shows the consistency constraint *CC* (difference between shared variables) corresponding to a sample unit (i.e. unit 25). Six curves are depicted each of which refers to a shared variable corresponding to unit 25 at the overlapping time intervals. Fig. 9*b* shows the rel index over the course of optimisation. The rel and *CC* values decrease as more iterations are carried out. The convergence measure rel is 0.0003 upon the convergence. We also implement the accelerated APP and present the results in Table 2. Although the convergence time is the same as the normal distributed algorithm, 93% of improvement is observed in rel upon the convergence. Fig. 9*b* depicts that the rel index obtained by the accelerated algorithm is always less than that for the normal algorithm. Note that, since several intertemporal constraints are heavily binding, CPU time of the centralised SCED increases by 43% and the distributed SCED takes two more iterations than Case 1. However, the distributed SCED still shows good performance and is 25% faster than the centralised SCED.

Case 4: Case 3 is reconsidered, and the suggested initialisation strategy is applied. The results are given in Table 2. Both normal and accelerated distributed SCED algorithms converge after 7 + 1 iterations within 11 s that is 27 and 45% less than the solution times of Case 3 and the centralised SCED, respectively. Since the intertemporal constraints between consecutive days are ignored in the initialisation step and ramping limits are binding heavily, the initialised solution is farther from the optimal point as compared with Case 2 in which the ramp limits are not binding heavily. Thus, the distributed SCED takes five iterations more than that for Case 2. Upon convergence, the rel values are less than those for Case 3. The accelerated method provides a smaller error compared to the normal algorithm.

Case 5: We consider ten contingency scenarios and compare the results with and without the suggested initialisation strategy. All other parameters are the same as in Case 2. Table 3 shows the

Table 2 Comparison of rel and simulation times obtained by A-APP, APP, and centralised SCED for Cases 3 and 4

Case no.	Algorithm	Iteration	rel	Time, s
	centralised	—	—	20
Case 3	APP	11	3×10^{-4}	15
	A-APP	11	2×10^{-5}	15
Case 4	APP + initialisation	7 + 1	2×10^{-6}	11
	A-APP + initialisation	7 + 1	6×10^{-7}	11

Table 3 Comparison of rel and simulation times obtained by A-APP, APP, and centralised SCED for Case 5

Algorithm	Iteration	rel	Time, s
centralised	—	—	59
APP	8	0.0005	31
a-APP	8	0.0002	31
APP + initialisation	2 + 1	1×10^{-7}	12
a-APP + initialisation	2 + 1	1×10^{-7}	12

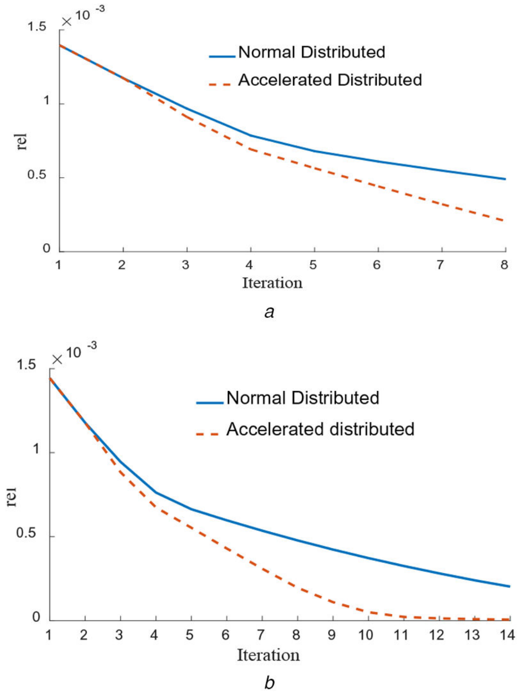

Fig. 10 *Rel index for normal and accelerated APP without initialisation obtained in*
(a) Case 5, and ***(b)*** Case 6

Table 4 Comparison of rel and simulation times obtained by A-APP, APP, and centralised SCED for Case 6

Algorithm	Iteration	rel	Time, s
centralised	—	—	70
APP	14	0.0002	54
a-APP	14	6×10^{-6}	54
APP + initialisation	9 + 1	2×10^{-6}	38
a-APP + initialisation	9 + 1	3×10^{-7}	38

comparison of the proposed A-APP with APP and the centralised SCED. The suggested initialisation strategy reduced the number of iterations (CPU time) from 8 (31 s) to 2 + 1 iterations (12 s). As observed in Fig. 10, the rel index obtained by the accelerated APP is always less than that for the normal APP. Increasing the number of credible contingencies from four to ten increases the computation time of the centralised SCED by 320%, while the distributed SCED is only 10% slower than that for Case 2. The distributed SCED takes 12 s to converge that is almost five times faster than the centralised SCED that takes 59 s. The difference between each pair of the shared variables and the relative error are almost zero upon convergence.

Case 6: This is similar to Case 5, but we decrease the permissible ramping of several units. Therefore, the difference between the shared variables after the initialisation step is much more than that in Case 5. As shown in Table 4, the distributed SCED with (without) the suggested initialisation strategy converges after 9 + 1 (14) iterations that is seven (six) iterations more than that for Case 5. The CPU time with (without) the suggested initialisation is 38 (54) s, which is 46% (30%) faster than the centralised method. As observed in Table 4 and Fig. 10, the accelerated APP provides a smaller error than the normal APP. In most simulations, upon convergence, the mismatch between each pair of shared variables is much smaller than 1%. Note that the power balance constraints are considered as hard constraints and are always satisfied regardless of the mismatch in generations. Considering the sign of mismatches, the summation of mismatches is zero at every interval.

Insight on selecting suitable multipliers: Although we have used the same multipliers for Cases 1–6 to have a fair comparison, this set of multipliers may not be the best choice for all cases. We can change the multipliers to obtain better results. In this section, we perform a sensitivity analysis and give an insight on how to initialise multipliers β, ρ, and γ. We examine the effect of multipliers and found suitable sets for Cases 1–4. By changing multipliers from 0.1 to 10, we plot the rel index versus the number of iterations required for convergence to determine the most suitable initial point for each case. Fig. 11 shows the simulation results for different multiplier values. The best solutions are highlighted by red colour in Fig. 11 and their corresponding multipliers are shown in Table 5. Cases 1 and 3 are more sensitive to the choice of initial values, while Cases 2 and 4 in which the suggested initialisation strategy is implemented are less sensitive to initial values. As expected, we have observed that selecting large multipliers potentially reduces the number of iterations; however, it increases error.

When a user wants to use the distributed SCED algorithm with no historical information, we recommend selecting small values for multipliers. The user will gradually obtain knowledge on suitable values of multipliers when more simulations are carried out. Evaluating the recorded results, the user can adjust the multipliers to enhance the solution performance for future simulations. In addition, it is proper to select ρ and γ equal to β [35].

5.2 472-Bus System

Case 7: We connect four IEEE 118-bus systems to make a large 472-bus system and compare changes in simulation times. Ten contingency scenarios are considered for each hour. The operation cost using both centralised and distributed SCED algorithms is \$44,376,003. The number of iterations, rel, and solution time using the normal APP and the proposed A-APP are compared with the

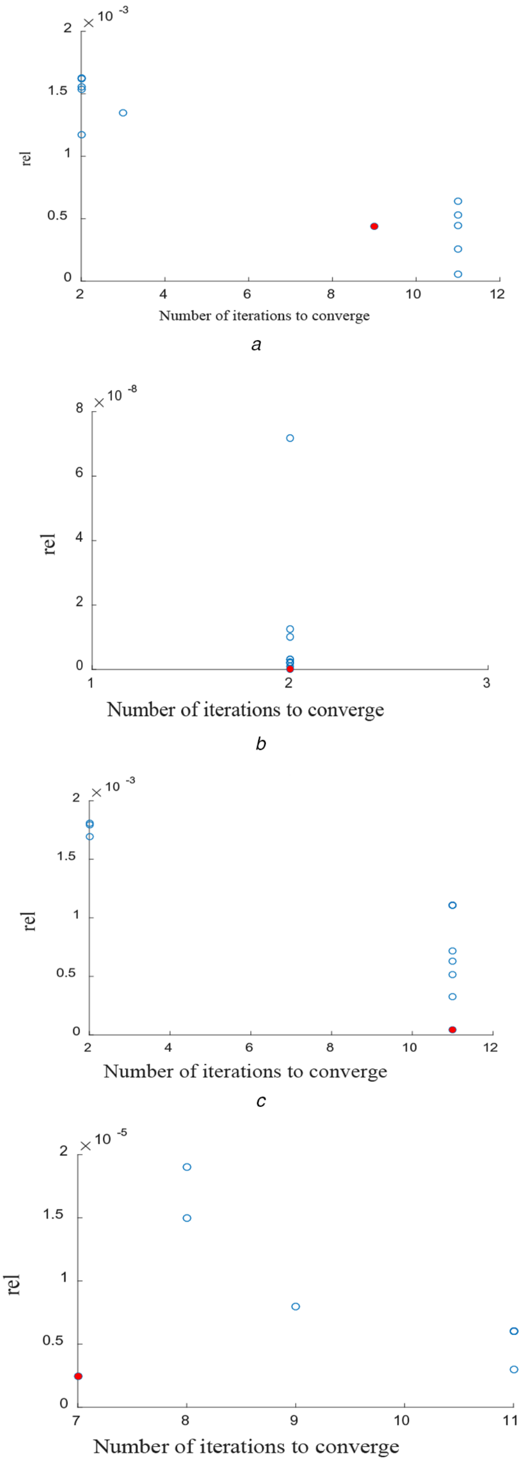

Fig. 11 *Relative error and the number of iterations to converge depending on the changes of multipliers \beta , ρ, and γ,*
(a) Case 1, ***(b)*** Case 2, ***(c)*** Case 3, and ***(d)*** Case 4

centralised SCED in Table 6. The convergence measure rel is almost zero upon convergence. The centralised SCED takes around 40 mi, but both normal and accelerated distributed algorithms converge after 2 min that are around 94% faster than the centralised algorithm.

Although the size of the system is four times larger than the 118-bus system, the solution time of centralised SCED is increased around 40 times compared to Case 5, which has the same loading.

Case 8: We manipulate ramping capability of generating units to make ramping limits of several units heavily binding in several hours and make more consistency constraints active. In this situation, the shared variables of neighbouring subproblems have a larger difference at the initialisation step, and the initial solution is farther from the feasible and optimal point. The results are summarised in Table 7. The operation cost obtained by the centralised SCED is $44,388,886. Although the distributed algorithm takes seven iterations more than Case 8, it still is 72% faster than the centralised SCED.

Table 5 Best multipliers for Cases 1–4

Case No.	β, ρ, γ	rel	Iteration
Case 1	0.2	4×10^{-4}	9
Case 2	1.25	1×10^{-10}	2
Case 3	0.1	4×10^{-5}	11
Case 4	0.1	1×10^{-8}	7

Table 6 Comparison of rel and simulation times obtained by A-APP, APP, and centralised SCED for Case 7

Algorithm	Iteration	rel	Time, s
centralised	—	—	2400
APP + initialisation	2 + 1	1×10^{-8}	140
a-APP + initialisation	2 + 1	1×10^{-8}	140

Table 7 Comparison of rel and simulation times obtained by A-APP, APP, and Centralised SCED for case 8

Algorithm	Iteration	rel	Time, s
centralised	—	—	1900
APP + initialisation	9 + 1	1×10^{-6}	532
a-APP + initialisation	9 + 1	1×10^{-7}	532

Table 8 Comparison of proposed distributed and centralised SCED approaches for 4720-bus system

Algorithm	Iterations	rel	Time, s
centralised	—	1 + 1	550
accelerated distributed	1 + 1	3×10^{-9}	141

5.3 4720-Bus System

Case 9: A day-ahead scheduling problem is considered for a 4720-bus test system. The centralised SCED takes 550 s to provide a cost of $64,736,891. The time horizon is decomposed into four subhorizons, each including six hours. As shown in Table 8, the proposed distributed SCED converges after 1 + 1 iterations (1 + 1 means that the algorithm takes one iteration to converge in addition to the one iteration needed for the initialisation step) within 141 s. The rel index is almost zero, and the solution time is improved by a factor around four.

Case 10: We have studied a network-constrained ED for this system to analyse the effect of the number of subproblems on the overall solution time. Fig. 12 shows the solution time variation versus the number of subproblems. In addition, Table 9 represents the number of iterations, rel values, and simulation times. A considerable time saving is achieved even if the time horizon is decomposed into a few sub-horizons. Increasing the number of sub-horizons increases the number of shared variables and the required iterations for A-APP to converge. Hence, the time saving is not significant if the number of sub-horizons goes beyond a certain number. Our simulations show that having four sub-horizons for this test system provides a good time saving and rel index. Although having eight subproblems provides a better time saving that having four subproblems, its rel index is higher.

6 Conclusion

In this paper, a decomposition strategy is proposed to divide the ramp-constrained SCED problem. Since the optimisation is decomposed over the scheduling time horizon, the proposed strategy is called horizontal time decomposition. The concept of

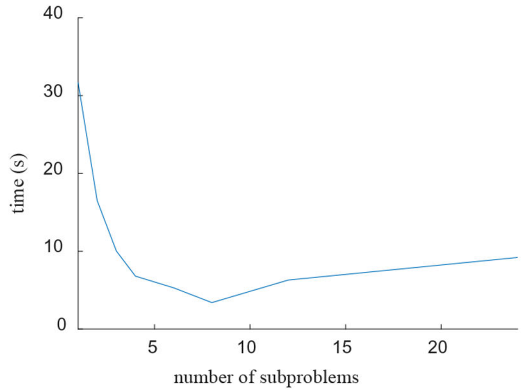

Fig. 12 *Time versus the number of subproblems for the 4720-bus system*

Table 9 Number of iterations, rel, and simulation times obtained for different number of subproblems

Number of Subproblems	Iteration	rel	Time, s
1	—	—	31.7
2	1 + 1	1×10^{-9}	16.5
3	1 + 1	1×10^{-9}	10.04
4	1 + 1	1×10^{-10}	6.8
6	2 + 1	1×10^{-6}	5.3
8	2 + 1	1×10^{-5}	3.4
12	6 + 1	2×10^{-5}	6.3
24	10 + 1	9×10^{-6}	9.2

overlapping time intervals is introduced to model interdependencies (coupling constraints) between the subproblems, which originate from intertemporal (or ramping) constraints of generating units. An accelerated APP and an initialisation strategy are proposed to enhance the convergence performance of the distributed SCED algorithm.

The simulation results show that the proposed algorithm reduces the computation time of SCED for the IEEE 118-bus system from 45 to 400% depending on the size of the considered problem. For cases with more contingency scenarios and small generation ramping capabilities, more time-saving is obtained using the distributed SCED instead of the centralised method. The simulation results on large test systems show that as the size of the problem (that depends on the size of the system and the number of contingencies) increases, the distributed algorithm shows better performance.

7 References

[1] Cain, M.B., O'Neill, R.P., Castillo, A.: '*History of optimal power flow and formulations*' (Federal Energy Regulatory Commission, USA, 2012), pp. 1–36

[2] Kargarian, A., Mohammadi, J., Guo, J., *et al.*: 'Toward distributed/decentralized DC optimal power flow implementation in future electric power systems', *IEEE Trans. Smart Grid*, 2018, **9**, (4), pp. 2574–2594

[3] Palomar, D.P., Chiang, M.: 'A tutorial on decomposition methods for network utility maximization', *IEEE J. Sel. Areas Commun.*, 2006, **24**, (8), pp. 1439–1451

[4] Molzahn, D.K., Dörfler, F., Sandberg, H., *et al.*: 'A survey of distributed optimization and control algorithms for electric power systems', *IEEE Trans. Smart Grid*, 2017, **8**, (6), pp. 2941–2962

[5] Kar, S., Hug, G., Mohammadi, J., *et al.*: 'Distributed state estimation and energy management in smart grids: a consensus + innovations approach', *IEEE J. Sel Top. Signal Process.*, 2014, **8**, (6), pp. 1022–1038

[6] Kargarian, A., Fu, Y., Li, Z.: 'Distributed security-constrained unit commitment for large-scale power systems', *IEEE Trans. Power Syst.*, 2015, **30**, (4), pp. 1925–1936

[7] Kargarian, A., Mehrtash, M., Falahati, B.: 'Decentralized implementation of unit commitment with analytical target cascading: a parallel approach', *IEEE Trans. Power Syst.*, 2018, **33**, (4), pp. 3981–3993

[8] Liu, J.-H., Chu, C.-C.: 'Iterative distributed algorithms for real-time available transfer capability assessment of multiarea power systems', *IEEE Trans. Smart Grid*, 2015, **6**, (5), pp. 2569–2578

[9] Baldick, R., Kim, B.H., Chase, C., *et al.*: 'A fast distributed implementation of optimal power flow', *IEEE Trans. Power Syst.*, 1999, **14**, (3), pp. 858–864

[10] Hur, D., Park, J.-K., Kim, B.H.: 'Evaluation of convergence rate in the auxiliary problem principle for distributed optimal power flow', *IEE Proc.-Gener. Transm. Distrib.*, 2002, **149**, (5), pp. 525–532

[11] Hu, J., Chen, M.Z., Cao, J., *et al.*: 'Coordinated active power dispatch for a microgrid via distributed lambda iteration', *IEEE J. Emerg. Sel Top. Circuits Syst.*, 2017, **7**, (2), pp. 250–261

[12] Cherukuri, A., Cortés, J.: 'Distributed generator coordination for initialization and anytime optimization in economic dispatch', *IEEE Trans. Control Netw. Syst.*, 2015, **2**, (3), pp. 226–237

[13] Feizollahi, M.J., Costley, M., Ahmed, S., *et al.*: 'Large-scale decentralized unit commitment', *International J. Electr. Power Energy Syst.*, 2015, **73**, pp. 97–106

[14] Boyd, S., Parikh, N., Chu, E., *et al.*: 'Distributed optimization and statistical learning via the alternating direction method of multipliers', *Found. Trends® in Mach. Learn.*, 2011, **3**, (1), pp. 1–122

[15] Conejo, A.J., Nogales, F.J., Prieto, F.J.: 'A decomposition procedure based on approximate Newton directions', *Math. Program.*, 2002, **93**, (3), pp. 495–515

[16] Cohen, G.: 'Auxiliary problem principle and decomposition of optimization problems', *J. Optim. Theory Appl.*, 1980, **32**, (3), pp. 277–305

[17] Mohammadi, A., Mehrtash, M., Kargarian, A.: 'Diagonal quadratic approximation for decentralized collaborative TSO + DSO optimal power flow', *IEEE Trans. Smart Grid*, 2019, **10**, (3), pp. 2358–2370

[18] Guo, F., Wen, C., Mao, J., *et al.*: 'Hierarchical decentralized optimization architecture for economic dispatch: a new approach for large-scale power system', *IEEE Trans. Ind. Inf.*, 2018, **14**, (2), pp. 523–534

[19] Lin, C., Wu, W., Chen, X., *et al.*: 'Decentralized dynamic economic dispatch for integrated transmission and active distribution networks using multi-parametric programming', *IEEE Trans. Smart Grid*, 2018, **9**, (5), pp. 4983–4993

[20] Amini, M.H., Jaddivada, R., Mishra, S., *et al.*: 'Distributed security constrained economic dispatch'. Innovative Smart Grid Technologies-Asia (ISGT ASIA), 2015, Bangkok, Thailand, 2015

[21] Xu, T., Wu, W., Sun, H., *et al.*: 'Fully distributed multi-area dynamic economic dispatch method with second-order convergence for active distribution networks', *IET Gener., Transm. Distrib.*, 2017, **11**, (16), pp. 3955–3965

[22] Chen, G., Zhao, Z.: 'Delay effects on consensus-based distributed economic dispatch algorithm in microgrid', *IEEE Trans. Power Syst.*, 2018, **33**, (1), pp. 602–612

[23] Li, P., Liu, Y., Xin, H., *et al.*: 'A robust distributed economic dispatch strategy of virtual power plant under cyber-attacks', *IEEE Trans. Ind. Inf.*, 2018, **14**, (10), pp. 4343–4352

[24] Federal Energy Regulatory Commission: 'Security constrained economic dispatch: definition, practices, issues and recommendations', A Report to Congress Regarding the Recommendations of Regional Joint Boards For The Study of Economic Dispatch Pursuant to Section. Vol. 223, 2006

[25] Wood, A.J., Wollenberg, B.F.: '*Power generation, operation, and control*' (John Wiley & Sons, Hoboken, NJ, USA, 2012)

[26] Low, S.H.: 'Convex relaxation of optimal power flow – part Ii: exactness', *IEEE Trans. Control Netw. Syst.*, 2014, **1**, (2), pp. 177–189

[27] Wu, H., Shahidehpour, M., Li, Z., *et al.*: 'Chance-constrained day-ahead scheduling in stochastic power system operation', *IEEE Trans. Power Syst.*, 2014, **29**, (4), pp. 1583–1591

[28] Bertsekas, D.P.: 'Nonlinear programming', *J. Oper. Res. Soc.*, 1997, **48**, (3), pp. 334–334

[29] Nesterov, Y.: 'A method of solving a convex programming problem with convergence rate O (1/K2)', in '*Soviet mathematics Doklady*' vol. 269, (Russia, 1983, pp. 543–547)

[30] Goldstein, T., O'Donoghue, B., Setzer, S., *et al.*: 'Fast alternating direction optimization methods', *SIAM J. Imaging Sci.*, 2014, **7**, (3), pp. 1588–1623

[31] Ruder, S.: 'An overview of gradient descent optimization algorithms'. arXiv preprint arXiv:1609.04747, 2016

[32] Nesterov, Y.E.: 'A method for solving the convex programming problem with convergence rate O ($1/K^2$)'. Dokl. Akad. Nauk SSSR, Russia, 1983

[33] Wang, Z., Shen, C., Liu, F., *et al.*: 'Chance-constrained economic dispatch with non-Gaussian correlated wind power uncertainty', *IEEE Trans. Power Syst.*, 2017, **32**, (6), pp. 4880–4893

[34] Lofberg, J.: 'Yalmip: a toolbox for modeling and optimization in Matlab'. 2004 IEEE Int. Symp. on Computer Aided Control Systems Design, Taipei, Taiwan, 2004

[35] Kim, B.H., Baldick, R.: 'Coarse-grained distributed optimal power flow', *IEEE Trans. Power Syst.*, 1997, **12**, (2), pp. 932–939