Look-Ahead SCOPF (LASCOPF) for Tracking Demand Variation via Auxiliary Proximal Message Passing (APMP) Algorithm ☆

Sambuddha Chakrabarti[a,*], Ross Baldick[b]

[a]*Division of Automatic Controls (Reglerteknik/RT), Department of Electrical Engineering (Institutionen för Systemteknik/ISY), Linköpings Universitet, Linköping 58183, Sweden*
[b]*Department of Electrical & Computer Engineering (ECE), The University of Texas at Austin, Austin, TX 78705, USA*

Abstract

In this paper, we will consider the Look-Ahead Security Constrained Optimal Power Flow (LASCOPF) problem looking forward multiple dispatch intervals, in which the load demand varies over dispatch intervals according to some forecast. We will consider the base-case and several contingency scenarios in the *upcoming* as well as in the *subsequent* dispatch intervals. We will formulate and solve the problem in a Model Predictive Control (MPC) paradigm. We will present the *Auxiliary Proximal Message Passing (APMP)* algorithm to solve this problem, which is a bi-layered decomposition-coordination type distributed algorithm, consisting of an outer *Auxiliary Problem Principle (APP)* layer and an inner *Proximal Message Passing (PMP)* layer. The *APP* part of the algorithm distributes the computation across several dispatch intervals and the *PMP* part performs the distributed computation within each of the dispatch interval across different devices (i.e. generators, transmission lines, loads) and nodes or nets. We will demonstrate the effectiveness of our method with a series of numerical simulations.

1. Introduction

In this paper, we will consider the $(N-1)$ *Security-Constrained Optimal Power Flow (SCOPF)* problem, in which devices (i.e. generators, transmission lines, loads etc.) are connected on the power network and there exists a set of *scenarios* — each corresponding to the failure of

---

*I am the corresponding author
Email addresses: `schakrabarti@utexas.edu` (Sambuddha Chakrabarti), `baldick@ece.utexas.edu` (Ross Baldick)

a particular transmission line — over which we must ensure optimal feasible operation of the network. But, instead of solving the SCOPF for the upcoming dispatch interval only, at the start of each such interval, we will also look forward several subsequent dispatch intervals, in which the load demands change according to a forecast. We will take into account the ramp rates of generators and operate the system such that the generation cost is minimized, subject to the constraints of satisfying the changing load demand, line power flow limit constraints, minimum and maximum generation constraint of each generator, enforced in each of the upcoming and subsequent dispatch intervals considered.

The goal is to minimize a composite cost function that includes the cost (and constraints) of nominal operation, as well as those associated with constraints on operation in any of the (adverse) scenarios, for multiple dispatch intervals. This results in a large optimization problem, since variables in the network, namely, real power injection and bus voltage phase angle, are repeated $|\mathcal{L}|$ times for each of the $|\Omega|$ dispatch intervals, where $|\mathcal{L}|$ is the number of contingencies. Therefore we present a distributed algorithm to solve this problem, whereby we break the problem into several smaller independent sub-problems and solve them in parallel, such that the only co-ordination required is local, through the Locational Marginal Prices (LMP) updates. Specifically, we use a suitably modified version of the message passing algorithm from [1] and [2] to solve this problem efficiently.

For simplicity, we consider only DC power flow in this paper. The extension to AC power flow, involves simply applying the AC-OPF model from [3], [4], [5], [6], [7], and [8] to each scenario and requiring that the phase angles of a given device are equal across all scenarios in the respective time periods.

In the previous works in [1], [9], and [2], Kraning *et al.*, Liu *et al.* and Chakrabarti *et al.* respectively applied the Proximal Message Passing Algorithm to solving the standard Static Optimal Power Flow (OPF) Problem to solving the $(N-1)$ Security Constrained OPF (SCOPF). In this paper, we extend this approach to solving a look-ahead dispatch problem, which considers the variation of load over a time horizon. There can be interesting and direct underpinnings of solving such problems, when it comes to design and allocation of Financial Transmission Rights (FTRs) under random outages, as well [10]. Our formulation for the LASCOPF is based on the Model Predictive Control (MPC) or Receding Horizon Control (RHC) methodology [11, 12, 13, 14, 15]. The rest of the paper is organized as follows: In section 2 we present a brief literature survey, after which, in section 3 we introduce the system of notations and present the conventional formulation of look-ahead SCOPF in the "Angles Represented" version, where the different real power flows on the Transmission Lines are represented in terms of the real powers injected at the buses and the voltage phase angles at the buses. In section 4, we apply the coarse-grained APP decomposition to split the problem into different dispatch intervals. In section 5, we reformulate all the preceding SCOPF problems (belonging to different dispatch intervals) into a different framework, the $\mathcal{DTN}$ (Devices-Terminals-Nets) formulation, which is particularly suitable for applying the ADMM (Alternating Direction Method of Multipliers) [16] based Proximal Message Passing algorithm to the problems. In section 6, we present the Proximal Message Passing fine-grained decomposition for the present problem. In section 7, we present the results of some simulation studies conducted on the IEEE test systems and we draw the concluding

2

remarks and point to future research in section 8.

2. Literature Survey and Related work

The Optimal Power Flow Problem is at the heart of Power Systems planning and operations. Initially formulated by Carpentier in [17], it has been studied for more than half a century, and has extensively been applied in the industry as well, including solving the electricity market scheduling and dispatch calculations by the ISOs/TSOs (which involves solving the DC-OPF) and applications by the market participants for planning using production-cost modeling systems, like Uplan [18]. Happ in [19] and Chowdhury & Rahman in [20] trace the historical development of OPF upto the 90s. A recent reference that provides a good summary of the historical development of the problem is Cain, O'Neill and Castillo [21]. The references cited there also provide good insights into formulation and modeling particularly, of the ACOPF. There has been recently a surge in the interest in ACOPF following the works of Bai *et al.* in [3] and Lavaei *et al.* in [4], [5], [6], [7], where the authors showed that the ACOPF problem can be solved accurately by a convex relaxation of the original problem into a Semidefinite Programming problem. In reference [22], the authors have presented the AC-OPF convex relaxations using Second Order Cone Programming (SOCP) for branch flow models, while reference [23] by Low provides an excellent and comprehensive guide to the current state of research in AC-OPF convex relaxation, making a special mention of the theory of chordal graphs, clique tree-decomposition, and SDP, as applied to power networks. Utlilizing these two ideas, recent references [24] and [25] mention applying distributed algorithms to AC-OPF, respectively, in the contexts of a distribution level microgrid and reactive power compensation. Reference [26] illustrates application of ADMM to sparse SDP for solving AC-OPF. In the recent reference [27], the authors presented a modified version of the ADMM, showing that a global optimum for the highly non-convex AC-OPF can be obtained by applying this algorithm. The recent trend of applying distributed optimization algorithms to power flow problem, in order to speed up the computation time has gained ample momentum already. Two of the recent papers, that provide very good and comprehensive comparisons between the performance of several different algorithms from solving the OPF and SCOPF problems are [28] and [29], where the authors mentioned of ADMM, Proximal Message passing (PMP), APP, Consensus+Innovation (C+I), Optimality Condition Decomposition (OCD) etc. algorithms.

The pioneering work on the Security Constrained OPF (SCOPF) was done by Stott *et al* in [30]. Some of the recent references include works by Chiang *et al.* [31], Phan *et al.* [32], Chakrabarti *et al.* [2] etc. The last two deal with application of distributed algorithms to the SCOPF problem. In [8] and in [9], the authors have applied ADMM to solve the ACOPF and SCOPF, but not to a look-ahead problem. Significant early works on ADMM method during the 70s and 80s were [33], [34], [35] etc. followed by work during the 90's which include [36], [37]. Combining these two fields gives rise to the Distributed Computational methods for OPF [38], [39]. The last reference provides a good comparison of the distributed methods till the end of 90s. Good references on Model Predictive Control (MPC) or Receding Horizon Control (RHC) are [11, 12, 13, 14, 15].

2.1. Contribution of the Present Work

While there has been some excellent work in the recent past on $(N-1)$ and $(N-k)$ SCOPFs as evidenced by references [40, 41, 42, 43, 44], however, there is a dearth of literature on multiple-dispatch interval SCOPF. Also, a vast majority of the academic literature present till date on the topic of SCOPF either solve a corrective control version of the SCOPF (by changing generator dispatch to relieve line congestion in the outaged cases) or they solve a chance constrained version. Moreover, in the academic body of literature, there is also a severe lack of materials, describing the software implementation for SCOPF and LASCOPF for the preventive control version. The main contribution of our work, in the light of the above, can be summarized as presenting the preventive control version of $(N-1)$ SCOPF for multiple dispatch intervals, as a part of which, we have also developed a software for implementing the mathematical model and the algorithms, which utilizes a two-layered distributed algorithm. In our present work, we will be solving the Look-Ahead Security Constrained Optimal Power Flow (LASCOPF) problem, formulated in a MPC/RHC format, using a combination of APP and ADMM-PMP algorithms, which we will call the APMP algorithm. As in all MPC/RHC formulation, the implicit assumption will be that, this problem for several look-ahead dispatch intervals is solved at the beginning of each dispatch interval, updating the forecasts, thereby reaching the best accuracy and ability to track the time-varying load demand.

2.2. Comparison of the Present Algorithm Against Other Methods

In the present method, that we are about to describe in this paper, the heart of the problem lies at solving the OPF problem using ADMM-PMP, while using APP to reach consensus among multiple different OPFs, that we solve. There are other methods also, to achieve the same goal. It's hard to say, which method is superior to which one. There are many factors and considerations, and the merit of a particular method depends on what factors are considered important, for a particular application. In the light of that, in the recent papers, [29] and [28] we have made comparisons against several different methods, which can be summarized as below:

The different algorithms compared are Analytical Target Cascading (ATC), ADMM (the pristine version), ADMM-PMP, APP, Optimality Condition Decomposition (OCD), and Consensus+Innovation (C+I) in the light of the following factors:

- *Presence of a central coordinator:* While ATC and ADMM do need central coordinators, ADMM-PMP, APP, OCD, and C+I do not need a central coordinator. This can speed up computations by reducing bottleneck of communication time with coordinator.

- *Computations Effort:* For a single iteration, ADMM-PMP and C+I has got much less computational effort as compared to all the other algorithms. However, ADMM-PMP needs many more iterations to converge. In our case, we stroke some compromise in the accuracy, in order to cut down on the ADMM-PMP iterations. In order to solve the LASCOPF, since we need to wrap the ADMM-PMP layer with the APP outer layers,

we cannot help reduce the net computational burden. However, it is considerably less, had any of the other methods been used.

- *Data Exchanged and Shared:* ADMM-PMP has a high volume of data exchanged and shared as compared to the other algorithms. However, given the previous point, combined with the fact that the data shared and exchanged volume of APP is quite low, the net communication time of APMP is moderate.

From the above comparison, we can conclude that APMP is a good candidate for distributed algorithm for solving LASCOPF problems. Certainly, for big networks and many contingency scenarios and dispatch intervals, we wouldn't prefer a centralized algorithm. Given below, is a table indicating comparison of APMP against a centralized solver implemented with GUROBI, as a benchmark solver. The percentage difference in optimal objective value is calculated as:

$\% \text{ Difference} = \frac{(\text{APMP Optimal Objective} - \text{Centralized Optimal Objective}) \times 100}{\text{Centralized Optimal Objective}}$

Table 1: Comparison of APMP against Centralized benchmark.

Type of the System	Percentage Difference
5 bus	0
14 bus	0.101128791
30 bus	0
57 bus	-0.003811375
118 bus	0
300 bus	-0.0001387213

3. Conventional Look-Ahead SCOPF Formulation

In this section, we will formulate the models for the look-ahead SCOPF in the conventional framework. First we will introduce the notations and symbols to be used for the rest of the paper, which are same as those introduced in [2] with a few additions.

3.1. Notations and Conventions

We have categorized the entities used in the subsequent formulations into five different groups: Sets, Elements, Index, Parameters, and Variables.

- *Sets*

    $\mathcal{D}$: Set of Devices The next three sets form partitions of the set of devices:

    $G \subseteq \mathcal{D}$: Set of Generators

    $\mathbb{T} \subseteq \mathcal{D}$: Set of Transmission Lines

$L \subseteq \mathcal{D}$: Set of Loads

$\mathcal{T}$: Set of Terminals

$\mathcal{N}$: Set of Nets or Buses or Nodes

$J(N_i)$: Set of buses/nodes/nets directly connected to node $N_i \in \mathcal{N}$

$\mathcal{L} = \{0, 1, 2, ..., |\mathcal{L}|\}$: Set of possible $(N-1)$ Contingencies. The element 0 indicates the base case

$\Omega = \{0, 1, 2, ..., |\Omega|\}$: Set of Dispatch intervals or, the net Dispatch Horizon under consideration. 1 indicates the upcoming dispatch interval under consideration, for which a dispatch decision will definitely be implemented. The rest, $\{2, 3, ..., |\Omega|\}$, are subsequent intervals in future. Hence, dispatch interval 0 is the current running one.

$\dagger$ will be used to denote the transpose of a vector or matrix.

- *Elements*

    t: Elements of $\mathcal{T}$

    g: Elements of G

    D: Elements of L

    T: Elements of $\mathbb{T}$

    N: Elements of $\mathcal{N}$

- *Indices*

    i, j: Nets

    k: Terminals

    q: Generators

    r: Transmission Lines

    d: Loads

    c: Contingencies

    τ: Dispatch Intervals

    ν: Iteration count for ADMM/PMP algorithm

    μ_{APP}: Iteration count for APP algorithm

- *Parameters*

    $R_{T_r}, X_{T_r}, Z_{T_r} = R_{T_r} + (\sqrt{-1})X_{T_r}, B_{T_r}$: Resistance, Reactance, Impedance, and Susceptance of the rth Transmission Line, respectively

    α, β, γ: APP tuning parameters, ρ: ADMM-PMP tuning parameter

$\alpha_{g_q}, \beta_{g_q}, \gamma_{g_q}$: Quadratic, Linear, and Constant Cost Co-efficients of the qth Generator

$C_{g_q}(.), f_{dev}(.)$ will be used to denote the cost function of the $g_q{}^{th}$ generator and that of a generic device, respectively, throughout. We will introduce the other cost functions in the appropriate sections

$I_{\leq}(.), I_{\geq}(.), I_{=}(.)$: Indicator functions such that $I_{\leq}(x) = 0$, if $0 \leq x$, and $= \infty$ otherwise, and similarly for $I_{\geq}(.)$ and $I_{=}(.)$

$\overline{P}_{g_q}, \underline{P}_{g_q}, \overline{R}_{g_q}, \underline{R}_{g_q}(=-\overline{R}_{g_q}$,usually$), \overline{L}_{T_r}$ denote the maximum and minimum generating limits of generators, maximum ramp-up and ramp-down limits of generators, and power carrying capacity of transmission lines respectively

- *Variables*

  P: The real power or, MW associated with different devices

  θ: The bus voltage phase angles associated with different nodes, as well as the devices

  λ: The lagrange multiplier or dual variable corresponding to the consensus among different power generation variables between the dispatch intervals

The following is the convention we follow in order to identify the associations of any particular variable to the sets:
$$\left(x_{d(p_1)(p_2)}^{(p_3)(p_4)} \right)^{(\nu)}.$$
In the above, x_d is a variable associated with a device-terminal or node/net-terminal, d. $(p_1)(p_2)$ and $(p_3)(p_4)$ indicate the computational units/agents, denoted, respectively, by a combination of p_1, p_2 and p_3, p_4, where a particular p_i may refer to either a dispatch interval or contingency scenario it is associated with. In this particular case, p_2 and p_3 might indicate, respectively, a contingency scenario and a dispatch interval. Together, the combination $(p_1)(p_2)$ denotes a particular computational agent linked to this particular combination. The above notation should be interpreted as the belief about the value of the variable, x_d associated or linked to the superscripted computational agent $((p_3)(p_4))$, as held by the subscripted computational agent $((p_1)(p_2))$ at the $\nu^{-\text{th}}$ iteration. Let's take a look at the next example, to make the meaning clearer.
$$\left(P_{N_t(c_1)(\tau_1)}^{(c_2)(\tau_2)} \right)^{(\nu)}.$$
The foregoing notation refers to the belief of the value of a variable P_{N_t} associated with a particular terminal, N_t, (which is indexed by the t) of either a net or a device for the contingency scenario indexed by c_2 during τ_2, as estimated by a computational agent linked to τ_1 and contingency scenario, c_1. Sometimes we will use the net number instead of the terminal number in the above convention, when we want to indicate several devices connected to a particular net. If it is part of an iterative algorithm, then the outermost superscript ν indicates the iteration count. Whenever a variable is boldface, one or more of the indices will be missing and that means the boldface variable is a vector each of whose components will

7

have all or some of the missing indices (the components themselves can be vectors or scalars). When the variable is not bold-face and still some of the indices are missing, that means it is a scalar and the missing indices are either irrelevant or their values are implied from the context. Also it is to be observed that since generators and loads are single terminal devices, it is not necessary to specify the terminals for these, unless absolutely required.

3.2. Look-Ahead Dispatch: Generalized Case of Demand Variation for Multi-Bus Case

In this paper we build on the static SCOPF problem formulation in [2]. We will develop the mathematical model for a look-ahead dispatch calculation that considers an upcoming and several subsequent dispatch intervals at the onset of each current dispatch interval and takes into account the possible variations of operating parameters across different scenarios represented in different subsequent dispatch intervals, so that at each interval the entire system is secure. At the end of each current dispatch interval, the calculation "rolls forward" and the whole look-ahead calculation is repeated with updated demand forecast, just as in a "Model Predictive Control (MPC)" or a "Receding Horizon Control (RHC)." The generalized version of this problem for a multi-time horizon, arbitrary network is as follows:

$$\min_{P_{g_q}^{(\tau)}, \theta} \sum_{\tau \in \Omega} \sum_{g_q \in G} C_{g_q}(P_{g_q}^{(\tau)}) \tag{1a}$$

Subject to $: \forall(c) \in \mathcal{L}, \forall \tau \in \Omega, \ \forall T_r \in \mathbb{T}$

$$P_{g_{q_{N_i}}}^{(\tau)} - P_{D_{d_{N_i}}}^{(\tau)} = \sum_{N_{\bar{i}} \in J(N_i)} B_{T_r}^{(c)}(\theta_{N_i}^{(c)(\tau)} - \theta_{N_{\bar{i}}}^{(c)(\tau)}); \ \forall N_i \in \mathcal{N} \tag{1b}$$

$$|B_{T_r}^{(c)}(\theta_{T_r t_1}^{(c)(\tau)} - \theta_{T_r t_2}^{(c)(\tau)})| \leq \overline{L}_{T_r}^{(c)}, \ \forall T_r \in \mathbb{T} \tag{1c}$$

$$\underline{R}_{g_q} \leq (P_{g_q}^{(\tau+1)} - P_{g_q}^{(\tau)}) \leq \overline{R}_{g_q} \ \forall g_q \in G \tag{1d}$$

In the optimization model (1), the objective function to be minimized, (1a) represents the total generation cost for all the generators in the system, summed over all the look-ahead dispatch intervals under consideration. The innermost summation is for all the generators and outermost summation is over all the future dispatch intervals. There are three sets of constraints. The first set of constraints (1b) describes the nodal power balance. The left-side of the equality represents the nodal power injection, which is the generator output, less the load connected to the particular node and the right-side of the equality represents the power flow along all the transmission lines that are connected to the node (according to the angles represented version of the DC power flow approximation). Inequality constraints (1c) represent the transmission lines' power flow limits for both the base case and each of the contingency scenarios for all the lines and all the dispatch intervals (the base case as well as the scenarios are indexed in the superscript with (c) and the dispatch intervals, by (τ)). As before, the left-side of the inequality represents the DC (or linearized) approximation of the line power flow and the right-side represents the limit. The last set of inequalities (1d) represent the generator ramping constraints, where it can be seen that the change in generator output from one dispatch interval to the other is bounded by the maximum and minimum ramping limits of the the generators.
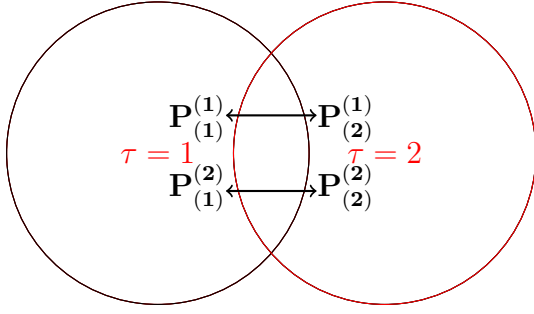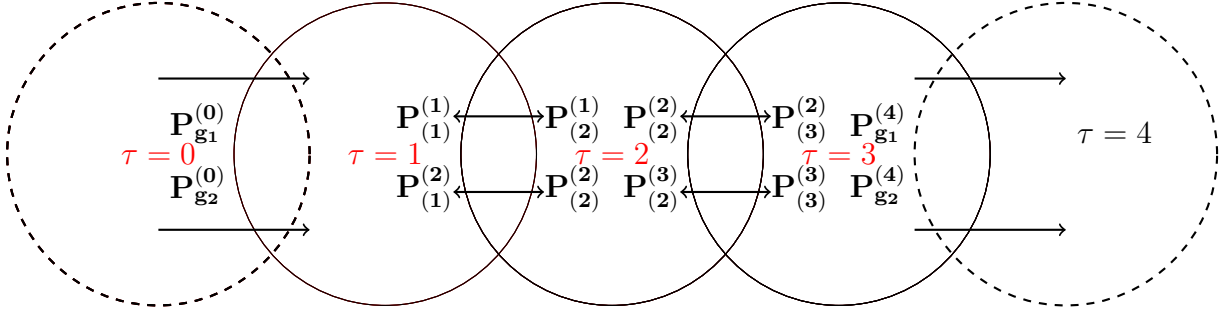
Figure 1: Schematic for APP Message Exchange.

Figure 2: Schematic for Look-Ahead SCOPF for Demand Variation

4. APMP Algorithm: The Coarse Grained Decomposition Component

In this section, we will provide the mathematical formulation of the Auxiliary Proximal Message Passing (APMP) algorithm, as applied to the LASCOPF problem. We will start by describing the APP based coarse-grained component of the algorithm, which will be applied to the LASCOPF problems, followed by the PMP fine-grained component, which decomposes the SCOPF problems (and the OPF problems) of each dispatch interval into the device level computations. The basic idea behind the APP based coarse-grained distributed component of the algorithm is depicted in figure 1, where the two circles correspond to different dispatch intervals and/or scenarios, across which the computation is split, and the arrows indicate the exchange of messages between those. The messages correspond to the beliefs of each circle or coarse grain about the values of power generations and/or injections/line flows within itself and also of those belonging to the neighboring circles.

4.1. Generalized Case of Demand Variation for Multi-Bus Case

Referring to the figure 2, the different overlapping circles here represent the different dispatch intervals, and with $\tau = 0$ and $\tau = 4$ representing respectively, the latest dispatch interval for which the LASCOPF problem is already solved & the results are known, and the dispatch interval following the concerned time horizon. In the coarse-grained decomposition, when $\tau = 1$, $P_{g_2(\tau)}^{(\tau-1)}$ and $P_{g_1(\tau)}^{(\tau-1)}$ are replaced by $P_{g_2}^{(0)}$ and $P_{g_1}^{(0)}$ respectively, which are the (known) MW outputs from the current dispatch interval. When $\tau = 3$, $P_{g_2(\tau)}^{(\tau+1)}$ and $P_{g_1(\tau)}^{(\tau+1)}$ are

9

replaced by $P_{g_2}^{(4)}$ and $P_{g_1}^{(4)}$ respectively, which are the last but one iterate values of the MW outputs from the last dispatch interval. We will now apply the Auxiliary Problem Principle (APP)[45], [46] to the above optimization problem to derive a set of expressions for the iterate updates. This reformulation is very similar in flavor to the ones presented previously in [38], [47], [48], [49] etc. We will state the coarse-grained APP based decomposition of the problem (the one for which the conventional formulation has been presented in equation (1)). We will distribute the SCOPF across each dispatch interval and then exchange messages regarding what a particular interval thinks about the optimal values of the decision variables for the intervals immediately preceding and succeeding it, eventually attempting to achieve a consensus between those. In the figure, the values shown at the two sides of the arrows are those "beliefs" among which we are trying to achieve consensus. We follow the same convention as introduced in section 3.1. As before, notice that each of the subproblems is very similar to the classical $(N-1)$ SCOPF problems, but with two important differences. First of all, each of these has ramp rate constraints, to account for the change in generator outputs over multiple intervals. Secondly, there are regularization terms added to the cost function for attaining consensus among different coarse grains about the values of the decision variables, which, in this case, are generator outputs. The second terms in the objective functions are the ones representing the proximity from previous iterates. The third and fourth terms in the objective functions of equations (2) and (4) and the third, fourth, fifth, and sixth terms in the objective function of equation (3) are the ones for attaining consensus among the coarse grains. The last terms in the objective functions of equations (2) and (4) and the last two terms in the objective function of equation (3) correspond to complementary slackness. Note that, in this case also, the real power generation variables are the only ones at which we wish to attain consensus through the application of APP. The bus voltage angles can be thought of as localized to the particular dispatch interval sub-problems, whose values at any iteration are determined by the power generation or injection profiles. We have split the LASCOPF problem into three types of sub-problems, each corresponding to a particular dispatch time interval.

First Dispatch Interval SCOPF:

$$\mathbf{P_{(1)}}^{(\mu_{APP}+1)} = \underset{\mathbf{P_{(1)}},\theta}{\operatorname{argmin}} \sum_{g_q \in G} C_{g_q}(P_{g_q(1)}^{(1)}) +$$

$$\frac{\beta}{2}||\mathbf{P_{(1)}} - \mathbf{P_{(1)}}^{(\mu_{APP})}||_2^2$$

$$+\gamma[\mathbf{P_{(1)}^{(1)}}^{\dagger}(\mathbf{P_{(1)}^{(1)}}^{(\mu_{APP})} - \mathbf{P_{(2)}^{(1)}}^{(\mu_{APP})}) +$$

$$\mathbf{P_{(1)}^{(2)}}^{\dagger}(\mathbf{P_{(1)}^{(2)}}^{(\mu_{APP})} - \mathbf{P_{(2)}^{(2)}}^{(\mu_{APP})})]$$

$$+\lambda_{\mathbf{1}}^{(\mu_{APP})\dagger}\mathbf{P_{(1)}^{(1)}} + \lambda_{\mathbf{2}}^{(\mu_{APP})\dagger}\mathbf{P_{(1)}^{(2)}} \tag{2a}$$

$$\textbf{Subject to}: \forall(c) \in \mathcal{L},\ \forall T_r \in \mathbb{T}$$

Power-Balance Constraints (Base-Case & Contingency):

$$P_{g_{q_{N_i}}(1)}^{(1)} - P_{D_{d_{N_i}}}^{(1)} = \sum_{N_{\bar{i}} \in J(N_i)} B_{T_r}^{(c)}(\theta_{N_i}^{(c)(1)} - \theta_{N_{\bar{i}}}^{(c)(1)});\ \forall N_i \in \mathcal{N} \tag{2b}$$

Flow Limit Constraints (Base-Case & Contingency):

$$|B^{(c)}_{T_r}(\theta^{(c)(1)}_{T_{rt_1}} - \theta^{(c)(1)}_{T_{rt_2}})| \le \overline{L}^{(c)}_{T_r}, \ \forall T_r \in \mathbb{T} \tag{2c}$$

Ramp-Rate Constraints:

$$\underline{R}_{g_q} \le P^{(2)}_{g_q(1)} - P^{(1)}_{g_q(1)} \le \overline{R}_{g_q}, \forall g_q \in G \tag{2d}$$

$$\underline{R}_{g_q} \le P^{(1)}_{g_q(1)} - P^{(0)}_{g_q} \le \overline{R}_{g_q}, \forall g_q \in G \tag{2e}$$

In the coarse-grained decomposed sub-problem (2), which is for the first look-ahead dispatch interval, the only differences from optimization model (1) are that, in the objective function, (2a) we don't have the summation over the dispatch intervals, for obvious reason, since the computation is split across the dispatch intervals. Also, we can see that, the objective function has some additional regularization terms. And as mentioned in the last paragraph, these are regularization terms added to the cost function for attaining consensus among different coarse grains about the values of the decision variables, which, in this case, are generator outputs. The second terms in the objective function is the one representing the proximity from previous iterate, $\mathbf{P_{(1)}}^{(\mu_{APP})}$. The third and fourth terms in the objective function (2a) are the ones for attaining consensus among the coarse grains. That is, in the third term, attaining consensus between $\mathbf{P}^{(1)(\mu_{APP})}_{(1)}$, which are the beliefs of interval 1 about its own generator outputs and $\mathbf{P}^{(1)(\mu_{APP})}_{(2)}$, which are the beliefs of interval 2 about the generator outputs of interval 1. The fourth term has a similar interpretation as to attaining consensus among interval 1's beliefs of generator outputs of interval 2 and interval 2's beliefs about its own. The last two terms in the objective function correspond to complementary slackness. Note that, in this case also, the real power generation variables are the only ones at which we wish to attain consensus through the application of APP. The bus voltage angles can be thought of as localized to the particular dispatch interval sub-problems, whose values at any iteration are determined by the power generation or injection profiles. The rest of the constraints corresponding to the nodal power balance, line flow limits, and generator ramping are exactly the same as those appearing in (1).

Intermediate Dispatch Interval $\tau \in \{2, 3, ..., |\Omega| - 1\}$ SCOPF:

$$\mathbf{P}_{(\tau)}^{(\mu_{APP}+1)} = \underset{\mathbf{P}_{(\tau)}, \theta}{\text{argmin}} \sum_{g_q \in G} C_{g_q}(P^{(\tau)}_{g_q(\tau)}) +$$

$$\frac{\beta}{2}||\mathbf{P}_{(\tau)} - \mathbf{P}_{(\tau)}^{(\mu_{APP})}||^2_2$$

$$+\gamma[\mathbf{P}^{(\tau-1)^\dagger}_{(\tau)}(\mathbf{P}^{(\tau-1)(\mu_{APP})}_{(\tau)} - \mathbf{P}^{(\tau-1)(\mu_{APP})}_{(\tau-1)}) +$$

$$\mathbf{P}^{(\tau)^\dagger}_{(\tau)}(\mathbf{P}^{(\tau)(\mu_{APP})}_{(\tau)} - \mathbf{P}^{(\tau)\ (\mu_{APP})}_{(\tau-1)})$$

$$+\mathbf{P}^{(\tau)^\dagger}_{(\tau)}(\mathbf{P}^{(\tau)(\mu_{APP})}_{(\tau)} - \mathbf{P}^{(\tau)\ (\mu_{APP})}_{(\tau+1)}) +$$

$$\mathbf{P}^{(\tau+1)^\dagger}_{(\tau)}(\mathbf{P}^{(\tau+1)(\mu_{APP})}_{(\tau)} - \mathbf{P}^{(\tau+1)(\mu_{APP})}_{(\tau+1)})]$$

$$+\lambda_{\mathbf{2\tau-1}}^{(\mu_{APP})\dagger}\mathbf{P}^{(\tau)}_{(\tau)} + \lambda_{\mathbf{2\tau}}^{(\mu_{APP})\dagger}\mathbf{P}^{(\tau+1)}_{(\tau)}$$

$$-\lambda_{2\tau-3}^{(\mu_{APP})\dagger}\mathbf{P}_{(\tau)}^{(\tau-1)} - \lambda_{2\tau-2}^{(\mu_{APP})\dagger}\mathbf{P}_{(\tau)}^{(\tau)} \tag{3a}$$

$$\textbf{Subject to} : \forall(c) \in \mathcal{L}, \ \forall T_r \in \mathbb{T}$$

Power-Balance Constraints (Base-Case & Contingency):

$$P_{g_{q_{N_i}}(\tau)}^{(\tau)} - P_{D_{d_{N_i}}}^{(\tau)} = \sum_{N_{\bar{i}} \in J(N_i)} B_{T_r}^{(c)}(\theta_{N_i}^{(c)(\tau)} - \theta_{N_{\bar{i}}}^{(c)(\tau)}); \ \forall N_i \in \mathcal{N} \tag{3b}$$

Flow Limit Constraints (Base-Case & Contingency):

$$|B_{T_r}^{(c)}(\theta_{T_{rt_1}}^{(c)(\tau)} - \theta_{T_{rt_2}}^{(c)(\tau)})| \leq \overline{L}_{T_r}^{(c)}, \ \forall T_r \in \mathbb{T} \tag{3c}$$

Ramp-Rate Constraints:

$$\underline{R}_{g_q} \leq P_{g_q(\tau)}^{(\tau+1)} - P_{g_q(\tau)}^{(\tau)} \leq \overline{R}_{g_q}, \forall g_q \in G \tag{3d}$$

$$\underline{R}_{g_q} \leq P_{g_q(\tau)}^{(\tau)} - P_{g_q(\tau)}^{(\tau-1)} \leq \overline{R}_{g_q}, \forall g_q \in G \tag{3e}$$

In the coarse-grained decomposed sub-problem (3), which are for the intermediate dispatch intervals, as opposed to the objective function for the optimization problem for the first dispatch interval, in (3a) there are four regularization terms after the one representing the proximity from previous iterate, $\mathbf{P}_{(\tau)}^{(\mu_{APP})}$. The third term is for attaining consensus between $\mathbf{P}_{(\tau)}^{(\tau-1)(\mu_{APP})}$, which are the beliefs of interval τ about the generator outputs of interval $\tau - 1$ and $\mathbf{P}_{(\tau-1)}^{(\tau-1)(\mu_{APP})}$, which are the beliefs of interval $\tau - 1$ about its own generator outputs. The fourth term has a similar interpretation as to attaining consensus among interval τ's beliefs of its own generator outputs and interval $(\tau - 1)$'s beliefs about generator outputs of interval τ, whereas the fifth term pertains to attaining consensus among interval τ's beliefs of its own generator outputs and interval $(\tau + 1)$'s beliefs about generator outputs of interval τ. The sixth term pertains to attaining consensus among the belief of interval τ about generator outputs of interval $\tau + 1$ and belief of interval $\tau + 1$ about its own generator outputs. The last four terms in the objective function correspond to complementary slackness. The rest of the constraints corresponding to the nodal power balance, line flow limits, and generator ramping are again, exactly the same as those appearing in (1).

Last Dispatch Interval, or Dispatch Interval $|\Omega|$ SCOPF:

$$\mathbf{P}_{(|\boldsymbol{\Omega}|)}^{(\mu_{APP}+1)} = \operatorname*{argmin}_{\mathbf{P}_{(|\boldsymbol{\Omega}|)}, \theta} \sum_{g_q \in G} C_{g_q}(P_{g_q(|\Omega|)}^{(|\Omega|)}) +$$

$$\frac{\beta}{2}||\mathbf{P}_{(|\boldsymbol{\Omega}|)} - \mathbf{P}_{(|\boldsymbol{\Omega}|)}^{(\mu_{APP})}||_2^2$$

$$+\gamma[\mathbf{P}_{(|\boldsymbol{\Omega}|)}^{(|\boldsymbol{\Omega}|-1)\dagger}(\mathbf{P}_{(|\boldsymbol{\Omega}|)}^{(|\boldsymbol{\Omega}|-1)(\mu_{APP})} - \mathbf{P}_{(|\boldsymbol{\Omega}|-1)}^{(|\boldsymbol{\Omega}|-1)(\mu_{APP})}) +$$

$$\mathbf{P}_{(|\boldsymbol{\Omega}|)}^{(|\boldsymbol{\Omega}|)\dagger}(\mathbf{P}_{(|\boldsymbol{\Omega}|)}^{(|\boldsymbol{\Omega}|)(\mu_{APP})} - \mathbf{P}_{(|\boldsymbol{\Omega}|-1)}^{(|\boldsymbol{\Omega}|)\ (\mu_{APP})})]$$

$$-\lambda_{2|\boldsymbol{\Omega}|-3}^{(\mu_{APP})\dagger}\mathbf{P}_{(|\boldsymbol{\Omega}|)}^{(|\boldsymbol{\Omega}|-1)} - \lambda_{2|\boldsymbol{\Omega}|-2}^{(\mu_{APP})\dagger}\mathbf{P}_{(|\boldsymbol{\Omega}|)}^{(|\boldsymbol{\Omega}|)} \tag{4a}$$

$$\textbf{Subject to} : \forall(c) \in \mathcal{L}, \ \forall T_r \in \mathbb{T}$$

Power-Balance Constraints (Base-Case & Contingency):

$$P_{g_{q_{N_i}}(|\Omega|)}^{(|\Omega|)} - P_{D_{d_{N_i}}}^{(|\Omega|)} = \sum_{N_{\bar{i}} \in J(N_i)} B_{T_r}^{(c)}(\theta_{N_i}^{(c)(|\Omega|)} - \theta_{N_{\bar{i}}}^{(c)(|\Omega|)});$$

$$\forall N_i \in \mathcal{N} \tag{4b}$$

Flow Limit Constraints (Base-Case & Contingency):

$$|B_{T_r}^{(c)}(\theta_{T_{r t_1}}^{(c)(|\Omega|)} - \theta_{T_{r t_2}}^{(c)(|\Omega|)})| \leq \overline{L}_{T_r}^{(c)}, \ \forall T_r \in \mathbb{T} \tag{4c}$$

Ramp-Rate Constraints:

$$\underline{R}_{g_q} \leq P_{g_q(|\Omega|)}^{(|\Omega|)(\mu_{APP})} - P_{g_q(|\Omega|)}^{(|\Omega|)} \leq \overline{R}_{g_q}, \forall g_q \in G \tag{4d}$$

$$\underline{R}_{g_q} \leq P_{g_q(|\Omega|)}^{(|\Omega|)} - P_{g_q(|\Omega|)}^{(|\Omega|-1)} \leq \overline{R}_{g_q}, \forall g_q \in G \tag{4e}$$

Dual Variable Updates :

$$\lambda_\tau^{(\mu_{APP}+1)} = \lambda_\tau^{(\mu_{APP})} + \alpha(\mathbf{P}_{(\tau)}^{(\tau)(\mu_{APP}+1)} - \mathbf{P}_{(\tau+1)}^{(\tau)}{}^{(\mu_{APP}+1)}) \tag{4f}$$

$$\lambda_{\tau+\mathbf{1}}^{(\mu_{APP}+1)} = \lambda_{\tau+\mathbf{1}}^{(\mu_{APP})} + \alpha(\mathbf{P}_{(\tau)}^{(\tau+\mathbf{1})(\mu_{APP}+1)} - \mathbf{P}_{(\tau+\mathbf{1})}^{(\tau+\mathbf{1})(\mu_{APP}+1)}) \tag{4g}$$

Lastly, in the coarse-grained decomposed sub-problem (4), which is for the last dispatch interval under consideration, in the objective function for the optimization problem (4a) the two regularization terms after the one representing the proximity from previous iterate, $\mathbf{P}_{(|\mathbf{\Omega}|)}^{(\mu_{APP})}$. The third term is for attaining consensus between $\mathbf{P}_{(|\mathbf{\Omega}|)}^{(|\mathbf{\Omega}|-1)(\mu_{APP})}$, which are the beliefs of interval $|\Omega|$ about the generator outputs of interval $|\Omega| - 1$ and $\mathbf{P}_{(|\mathbf{\Omega}|-1)}^{(|\mathbf{\Omega}|-1)(\mu_{APP})}$, which are the beliefs of interval $|\Omega| - 1$ about its own generator outputs. The fourth term has a similar interpretation as to attaining consensus among interval $|\Omega|$'s beliefs of its own generator outputs and interval $(|\Omega|-1)$'s beliefs about generator outputs of interval $|\Omega|$. The last two terms in the objective function correspond to complementary slackness. The rest of the constraints corresponding to the nodal power balance, line flow limits, and generator ramping are again, exactly the same as those appearing in (1). Equations (4f) and (4g) pertain to updates of the dual variables or the Lagrange multipliers, λs corresponding to the different power generation consensus terms described above. The lack of consensus gives the step direction, while α is the step length, as can be seen above.

We have presented a coarse grained parallelization of the look-ahead SCOPF to cope with demand variation, where the entire problem is decomposed across the different dispatch time intervals. Subsequently, we will present the ADMM-Proximal Message Passing based fine grained decomposition and parallelization of the problem.

5. $\mathcal{DTN}$ Reformulations of the SCOPF Problems

In this section we carry out the reformulation of the SCOPF models that we presented in the last section within each APP coarse grain, in order for us to be able to solve those problems by the Proximal Message Passing method. In order to simplify the presentation and also better clarify the meaning of each term, we will group the terms of the objective into four different categories. We will define them and write down the expressions pertaining

to the SCOPF in the coarse grain corresponding to equation (3). These are:

1) Cost of Generation ($C(P)$): This term consists of the actual total cost of generating real power by the different generators as well as the indicator functions corresponding to the lower and upper generating limits of the different generators. For this term, the real power generated is always considered at the base case. This term is given as:

$$C(\mathbf{P}^{(0)(\tau)}) = \sum_{t_k \in g_q \cap \mathcal{T}, q=1}^{|G|} \left(C_{g_q}(P_{g_{q_{t_k}}(\tau)}^{(0)(\tau)}) + I_{\leq}(\overline{P}_{g_q} - P_{g_{q_{t_k}}(\tau)}^{(0)(\tau)}) + I_{\leq}(P_{g_{q_{t_k}}(\tau)}^{(0)(\tau)} - \underline{P}_{g_q}) + \frac{\beta}{2} \left[(P_{g_{q_{t_k}}(\tau)}^{(0)(\tau)} - \right. \right.$$

$$\left. P_{g_{q_{t_k}}(\tau)}^{(0)(\tau)(\mu_{APP})})^2 + \sum_{s=-1,1}(P_{g_{q_{t_k}}(\tau)}^{(0)(\tau+s)} - P_{g_{q_{t_k}}(\tau)}^{(0)(\tau+s)(\mu_{APP})})^2 \right] + \gamma \left[\sum_{s=-1,1}(P_{g_{q_{t_k}}(\tau)}^{(0)(\tau)}(P_{g_{q_{t_k}}(\tau)}^{(0)(\tau)(\mu_{APP})} - \right.$$

$$\left. P_{g_{q_{t_k}}(0)(\tau+s)}^{(0)(\tau)(\mu_{APP})}) + P_{g_{q_{t_k}}(\tau)}^{(0)(\tau+s)}(P_{g_{q_{t_k}}(\tau)}^{(0)(\tau+s)(\mu_{APP})} - P_{g_{q_{t_k}}(0)(\tau+s)}^{(0)(\tau+s)(\mu_{APP})})) \right] + (\lambda_{g_q 2(\tau-1)}^{(\mu_{APP})} - \lambda_{g_q(2\tau-2)}^{(\mu_{APP})})P_{g_{q_{t_k}}(\tau)}^{(0)(\tau)} +$$

$$\left. \lambda_{g_q(2\tau)}^{(\mu_{APP})} P_{g_{q_{t_k}}(\tau)}^{(0)(\tau+1)} - \lambda_{g_q(2\tau-3)}^{(\mu_{APP})} P_{g_{q_{t_k}}(\tau)}^{(0)(\tau-1)} \right)$$

For the sake of brevity, we are presenting here the terms corresponding to line flow limit constraint ($F(P^{(c)(\tau)})$), power-angle relation ($\chi(P^{(c)(\tau)}, \theta^{(c)(\tau)})$), and ramp constraint ($\Delta(P^{(0)(\tau)})$) in the most general and condensed forms. The details of these can be found in [2] and [50].

2) Line Flow Limit Constraint ($F(P)$): This term consists of the sum of the indicator functions corresponding to enforcing that real power flows on the lines are less than the maximum allowed, both at the base-case as well as under different contingencies. This term can be written as:

$$F(P^{(c)(\tau)}) = \sum_{(c) \in \mathcal{L}} \sum_{T_r \in \mathbb{T}} \sum_{t_k \in T_r \cap \mathcal{T}} I_{\leq}(\overline{L}_{T_r}^{(c)} - |P_{T_r t_k}^{(c)(\tau)}|)$$

3) Power-Angle Relation ($\chi(P, \theta)$): This term consists of the sum of the indicator functions corresponding to the relation of the power flow at each end of the lines and the voltage phase angles at the two ends, both at the base-case and the contingencies. This term can be written as:

$$\chi(P^{(c)(\tau)}, \theta^{(c)(\tau)}) = \sum_{(c) \in \mathcal{L}} \sum_{T_r \in \mathbb{T}} \sum_{t_k, t_{k'} \in T_r \cap \mathcal{T}} I_{=}\left(P_{T_r t_k}^{(c)(\tau)} + \frac{\theta_{T_r t_k}^{(c)(\tau)} - \theta_{T_r t_{k'}}^{(c)(\tau)}}{X_{T_r}^{(c)}} \right)$$

4) Ramp Constraint ($\Delta(P)$): This term corresponds to the change of power output of generator from one time period to another and the maximum rate at which it can go up or down. This term can be written as:

$$\Delta(P^{(0)(\tau)}) = \sum_{s=-1,1} \sum_{t_k \in g_q \cap \mathcal{T}, q=1}^{|G|} (I_{\leq}(\overline{R}_{g_q} - P_{g_{q_{t_k}}(\tau)}^{(0)(\tau+s)} + P_{g_{q_{t_k}}(\tau)}^{(0)(\tau)}) + I_{\leq}(P_{g_{q_{t_k}}(\tau)}^{(0)(\tau+s)} - P_{g_{q_{t_k}}(\tau)}^{(0)(\tau)} - \underline{R}_{g_q}))$$

The reformulated equations for the problem stated in equations (2)-(4) are as follows:

$$\min_{P_{t_k}^{(c)(\tau)}, \theta_{t_k}^{(c)(\tau)}} f(P) = C(P^{(0)(\tau)}) + F(P^{(c)(\tau)}) +$$

$$\chi(P^{(c)(\tau)}, \theta^{(c)(\tau)}) + \Delta(P^{(0)(\tau)}) \tag{5a}$$

$$\text{Subject to: } \hat{P}_{N_{i t_k}}^{(c)(\tau)} = 0, \tilde{\theta}_{N_{i t_k}}^{(c)(\tau)} = 0,$$

$$\forall N_i \in \mathcal{N}, \forall t_k \in \mathcal{T}, \forall (c) \in \mathcal{L} \tag{5b}$$

In the above set of equations, $\hat{P}_{N_{i t_k}}^{(c)(\tau)}$ is the average node power injection and $\tilde{\theta}_{N_{i t_k}}^{(c)(\tau)}$ is the deviation of the voltage phase angle of each device connected to a particular node from the

average node voltage phase angle. Readers are referred to [2], [1] for more information on these. Hence, constraints stated in equation (5b) represent the Kirchhoff's law for node power balance and angle consistency.

6. APMP Algorithm: The Fine Grained Decomposition Component

In this section we state the equations for the Proximal Message Passing algorithm based fine-grained decomposition of the above problem.

6.1. Proximal Message Passing for the Look-Ahead Dispatch: Generalized Case of Demand Variation for the Multi Bus-Case

From equation (5), the slightly reformulated $\mathcal{DTN}$ equations are as follows:

$$
\begin{aligned}
&\min_{P_{t_k}^{(c)}, \theta_{t_k}^{(c)}} C(P^{(0)(\tau)}) + F(P^{(c)(\tau)}) + \chi(P^{(c)(\tau)}, \theta^{(c)(\tau)}) + \\
&\Delta(P^{(0)(\tau)}) + \sum_{(c)\in\mathcal{L}} \sum_{N_i\in\mathcal{N}} (\hat{I}(z_{N_i t_k}^{(c)(\tau)}) + \tilde{I}(\xi_{N_i t_k}^{(c)(\tau)})) \\
&\text{Subject to: } P_{t_k}^{(c)(\tau)} = z_{t_k}^{(c)(\tau)}, \theta_{t_k}^{(c)(\tau)} = \xi_{t_k}^{(c)(\tau)}, \\
&\forall N_i \in \mathcal{N}, \forall t_k \in \mathcal{T}, \forall(c) \in \mathcal{L}
\end{aligned}
\tag{6}
$$

where $\hat{I}(z_{N_i t_k})$ and $\tilde{I}(\xi_{N_i t_k})$ are indicator functions of the sets $\{z_{t_k} | \hat{z}_{N_i t_k} = 0\}$ and $\{\xi_{t_k} | \tilde{\xi}_{N_i t_k} = 0\}$ respectively. Again, for the sake of brevity, we will skip writing down the iterates for the transmission lines and loads (which are simple projections), and make a mention that they are the same as the ones presented in references [2] and [50].

6.1.1. Iterates for Generators

The iterates for generators consist of the update equations for the real power output and voltage-phase angles of the generator terminals for both the base case and the different $(N-1)$ contingency scenarios and are as follows:

$$
\begin{aligned}
&(P_{g_q t_k}^{(0)(\tau)(\nu+1)}, \theta_{g_q t_k}^{(c)(\tau)(\nu+1)}) \\
&= \operatorname{argmin}_{P_{g_q t_k}^{(0)(\tau)}, \theta_{g_q t_k}^{(c)(\tau)}} [C_{g_q}(P_{g_{q t_k}(\tau)}^{(0)(\tau)}) + I_{\le}(\overline{P}_{g_q} - P_{g_{q t_k}(\tau)}^{(0)(\tau)}) \\
&+ I_{\le}(P_{g_{q t_k}(\tau)}^{(0)(\tau)} - \underline{P}_{g_q}) + \frac{\beta}{2}[(P_{g_{q t_k}(\tau)}^{(0)(\tau)} - P_{g_{q t_k}(\tau)}^{(0)(\tau)(\mu_{APP})})^2 \\
&+ \sum_{s=-1,1}(P_{g_{q t_k}(\tau)}^{(0)(\tau+s)} - P_{g_{q t_k}(\tau)}^{(0)(\tau+s)(\mu_{APP})})^2] + \\
&\gamma[\sum_{s=-1,1}(P_{g_{q t_k}(\tau)}^{(0)(\tau)}(P_{g_{q t_k}(\tau)}^{(0)(\tau)(\mu_{APP})} - P_{g_{q t_k}(0)(\tau+s)}^{(0)(\tau)(\mu_{APP})}) \\
&+ P_{g_{q t_k}(\tau)}^{(0)(\tau+s)}(P_{g_{q t_k}(\tau)}^{(0)(\tau+s)(\mu_{APP})} - P_{g_{q t_k}(0)(\tau+s)}^{(0)(\tau+s)(\mu_{APP})}))] + \\
&(\lambda_{g_q 2(\tau-1)}^{(\mu_{APP})} - \lambda_{g_q(2\tau-2)}^{(\mu_{APP})})P_{g_{q t_k}(\tau)}^{(0)(\tau)} + \lambda_{g_q(2\tau)}^{(\mu_{APP})} P_{g_{q t_k}(\tau)}^{(0)(\tau+1)} - \\
&\lambda_{g_q(2\tau-3)}^{(\mu_{APP})} P_{g_{q t_k}(\tau)}^{(0)(\tau-1)} + \Delta(P^{(0)(\tau)}) + \\
&\sum_{(c)\in\mathcal{L}}(\frac{\rho}{2})(\|P_{g_{q t_k}}^{(0)(\tau)} - z_{g_q t_k}^{(c)(\tau)(\nu)} + u_{g_q t_k}^{(c)(\tau)(\nu)}\|_2^2 \\
&+ \|\theta_{g_q t_k}^{(c)(\tau)} - \xi_{g_q t_k}^{(c)(\tau)(\nu)} + v_{g_q t_k}^{(c)(\tau)(\nu)}\|_2^2)], \\
&\forall g_q \in G, \tau \in \Omega, t_k \in \mathcal{T} \cap G
\end{aligned}
$$

6.1.2. Iterates for Nets

We are writing here just the analytical forms already derived in [1].

$$\forall N_i \in \mathcal{N}, \forall t_k \in \mathcal{T} \cap N_i, \forall(c) \in \mathcal{L}, \tau \in \Omega$$

$$z_{N_i t_k}^{(c)(\tau)(\nu+1)}$$
$$= u_{N_i t_k}^{(c)(\tau)(\nu)} + P_{N_i t_k}^{(c)(\tau)(\nu+1)} - \hat{u}_{N_i t_k}^{(c)(\tau)(\nu)} - \hat{P}_{N_i t_k}^{(c)(\tau)(\nu+1)}$$
$$\xi_{N_i t_k}^{(c)(\tau)(\nu+1)} = \hat{v}_{N_i t_k}^{(c)(\tau)(\nu)} + \hat{\theta}_{N_i t_k}^{(c)(\tau)(\nu+1)}$$
$$u_{N_i t_k}^{(c)(\tau)(\nu+1)} = u_{N_i t_k}^{(c)(\tau)(\nu)} + (P_{N_i t_k}^{(c)(\tau)(\nu+1)} - z_{N_i t_k}^{(c)(\tau)(\nu+1)})$$
$$v_{N_i t_k}^{(c)(\tau)(\nu+1)} = v_{N_i t_k}^{(c)(\tau)(\nu)} + (\theta_{N_i t_k}^{(c)(\tau)(\nu+1)} - \xi_{N_i t_k}^{(c)(\tau)(\nu+1)})$$

Note that, not only do all the devices update their variables in parallel, but also, except the generators, all devices have associated with them the base-case and the contingency scenarios in each dispatch interval, each of which in turn update their respective variables in parallel as well. For the generators, the present dispatch interval and the next one are related through the ramp rate constraint ($\Delta(.)$ function). Then all the nets and the base-case/contingency scenarios associated with them update the first two set of variables in parallel and then update the next two in parallel. The proximal function for a function g is given by

$$\mathbf{prox}_{g,\rho}(\mathbf{v}) = \underset{\mathbf{x}}{\operatorname{argmin}}(g(\mathbf{x}) + (\rho/2)\|\mathbf{x} - \mathbf{v}\|_2^2).$$

For this case, the prox messages and the Proximal Message Passing Algorithm is as follows:

$$1.(P_{g_q t_k}^{(0)(\tau)(\nu+1)}, \theta_{g_q t_k}^{(c)(\tau)(\nu+1)})$$
$$= \mathbf{prox}_{C(P)+\Delta(P),\rho}(P_{g_q t_k}^{(0)(\tau)(\nu)} - \hat{P}_{g_q t_k}^{(c)(\tau)(\nu)} - u_{g_q t_k}^{(c)(\tau)(\nu)},$$
$$\hat{v}_{g_q t_k}^{(c)(\tau)(\nu-1)} + \hat{\theta}_{g_q t_k}^{(c)(\tau)(\nu)} - v_{g_q t_k}^{(c)(\tau)(\nu)}),$$
$$\forall g_q \in G, \forall(c) \in \mathcal{L}, \forall \tau \in \Omega$$

$$2.(P_{T_r t_k}^{(c)(\tau)(\nu+1)}, \theta_{T_r t_k}^{(c)(\tau)(\nu+1)}, P_{T_r t_{k'}}^{(c)(\tau)(\nu+1)}, \theta_{T_r t_{k'}}^{(c)(\tau)(\nu+1)})$$
$$= \mathbf{prox}_{F(P)+\chi(P,\theta),\rho}(P_{T_r t_k}^{(c)(\tau)(\nu)} - \hat{P}_{T_r t_k}^{(c)(\tau)(\nu)} - u_{T_r t_k}^{(c)(\tau)(\nu)},$$
$$\hat{v}_{T_r t_k}^{(c)(\tau)(\nu-1)} + \hat{\theta}_{T_r t_k}^{(c)(\tau)(\nu)} - v_{T_r t_k}^{(c)(\tau)(\nu)}),$$
$$\forall T_r \in \mathbb{T}, \forall(c) \in \mathcal{L}, \forall \tau \in \Omega$$

$$3.(P_{D_d t_k}^{(c)(\tau)(\nu+1)}, \theta_{D_d t_k}^{(c)(\tau)(\nu+1)})$$
$$= \mathbf{prox}_{-D,\rho}(\hat{v}_{D_d t_k}^{(c)(\tau)(\nu-1)} + \hat{\theta}_{D_d t_k}^{(c)(\tau)(\nu)} - v_{D_d t_k}^{(c)(\tau)(\nu)}),$$
$$\forall D_d \in L, \forall(c) \in \mathcal{L}, \forall \tau \in \Omega$$

$$4.u_{N_i t_k}^{(c)(\tau)(\nu+1)} = u_{N_i t_k}^{(c)(\tau)(\nu)} + \hat{P}_{N_i t_k}^{(c)(\tau)(\nu+1)},$$
$$\forall N_i \in \mathcal{N}, \forall(c) \in \mathcal{L}, \forall \tau \in \Omega$$

$$5.v_{N_i t_k}^{(c)(\tau)(\nu+1)} = \tilde{v}_{N_i t_k}^{(c)(\tau)(\nu)} + \tilde{\theta}_{N_i t_k}^{(c)(\tau)(\nu+1)},$$
$$\forall N_i \in \mathcal{N}, \forall(c) \in \mathcal{L}, \forall \tau \in \Omega$$
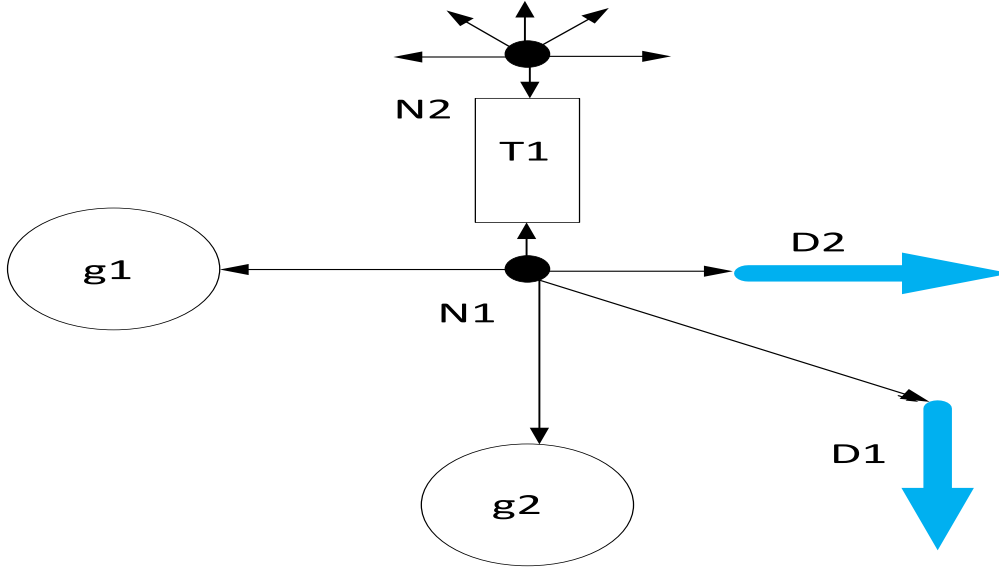
Figure 3: Broadcast: Calculation of Prox Function for Devices

In the above, steps 1, 2, and 3 consist of the concurrent calculation of prox functions by the different devices, independently, which happens after the most recently updated values of the dual variables are "broadcast" to the devices from the nodes, as shown in figure 3, in which, we have shown two generators (g_1, g_2), two loads (D_1, D_2), one transmission line (T_1), and two nodes (N_1, N_2). The direction of the arrows show the direction of conveying the most recently updated values of the dual variables. The calculations of steps 4 and 5, which refer to updating of the dual variables corresponding to nodal power balance and angle consistency, respectively follow the above-mentioned prox-function calculation by the devices. Once the devices finish calculating their respective prox-functions, the most recently updated decision variable values are sent back to the respective nodes, through a process called "gather," as shown in figure 4 The ADMM based PMP can be thought of as a recurring sequence of such broadcast-gather operation until the convergence criteria is satisfied, as represented in figure 5 for a generic set of several devices and nodes. The convergence criteria for the ADMM-PMP iterations is given as:

Stopping criterion for ADMM-PMP: We can define primal and dual residuals $\mathbf{r}^{(\nu)} \in \mathbb{R}^{|\mathcal{N}|+|\mathcal{T}|}$ and $\mathbf{s}^{(\nu)} \in \mathbb{R}^{|\mathcal{N}|+|\mathcal{T}|}$, respectively (at the end of ADMM-PMP iteration ν, for the
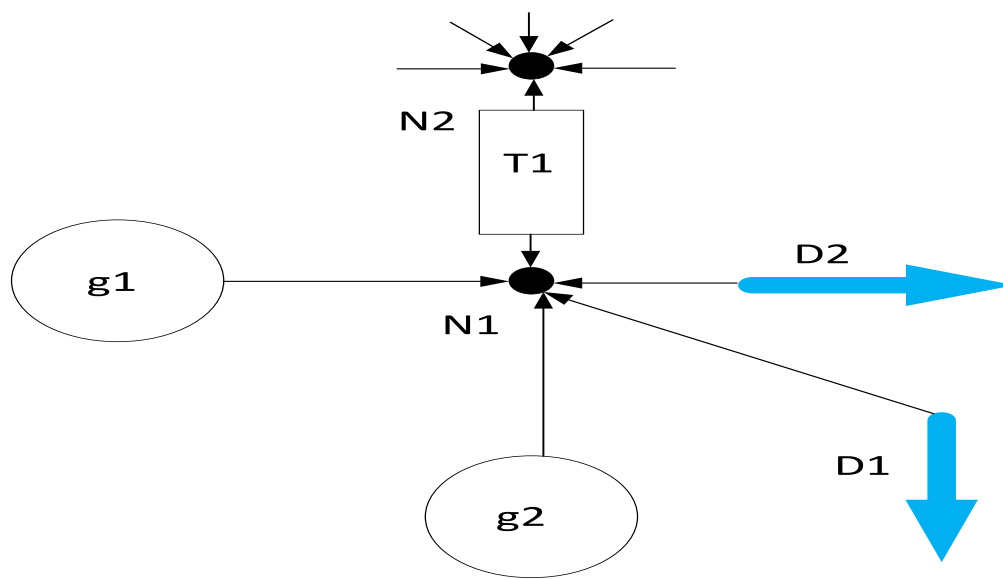
Figure 4: Gather: Updating of Dual variables by Nodes

PMP algorithm as:

$$\mathbf{r}^{(\nu)} = \left(\hat{\mathbf{P}}^{(\nu)}, \tilde{\boldsymbol{\Theta}}^{(\nu)} \right), \mathbf{s}^{(\nu)} = \rho \left(((\mathbf{P}^{(\nu)} - \mathbb{A} \times \hat{\mathbf{P}}^{(\nu)}) - (\mathbf{P}^{(\nu-1)} - \mathbb{A} \times \hat{\mathbf{P}}^{(\nu-1)})), (\hat{\boldsymbol{\Theta}}^{(\nu)} - \hat{\boldsymbol{\Theta}}^{(\nu-1)}) \right).$$

Where

- $\mathbb{A} \in \mathbb{R}^{|\mathcal{T}| \times |\mathcal{N}|}$ is the terminal to node incidence matrix, with

  - $\mathbb{A}(t_k, N_i) = 1$ if terminal t_k is connected to node N_i
  - otherwise, $\mathbb{A}(t_k, N_i) = 0$

- $\mathbf{P}^{(\nu)} \in \mathbb{R}^{|\mathcal{T}|}$ is a vector of the real power iterates of all the devices-terminals (generators, transmission lines, and loads) connected to a particular node, at the end of the $\nu^{\text{-th}}$ ADMM-PMP iteration.

- $\hat{\mathbf{P}}^{(\nu)} \in \mathbb{R}^{|\mathcal{N}|}$ is a vector of the real power nodal average injections (i.e. real power iterates of all the devices (generators, transmission lines, and loads) connected to a particular node, divided by the number of devices) at the end of the $\nu^{\text{-th}}$ ADMM-PMP iteration.

- $\tilde{\boldsymbol{\Theta}}^{(\nu)} \in \mathbb{R}^{|\mathcal{T}|}$ is the difference between the voltage phase angle iterate of each device, connected to a particular node and the average voltage phase angle at the node, at the end of the $\nu^{\text{-th}}$ ADMM-PMP iteration.

- $\hat{\boldsymbol{\Theta}}^{(\nu)} \in \mathbb{R}^{|\mathcal{N}|}$ is a vector of the voltage phase angle nodal average (i.e. voltage phase angle iterates of all the devices (generators, transmission lines, and loads) connected to a particular node, divided by the number of devices) at the end of the $\nu^{\text{-th}}$ ADMM-PMP iteration.

A simple terminating criterion for prox-project message passing is when

$$\|\mathbf{r}^{(\nu)}\|_2 \le \epsilon^{\text{pri}}, \qquad \|\mathbf{s}^{(\nu)}\|_2 \le \epsilon^{\text{dual}},$$

where ϵ^{pri} and ϵ^{dual} are, respectively, primal and dual tolerances and the norms are second norms. The entire APMP algorithm can be pictorially represented by the three flowcharts, shown in Figures 6, 7, 8, which respectively show the algorithmic steps for the outermost APP for attaining consensus among the beliefs about power generations about different intervals, inner APP for attaining consensus among the base-case generation value beliefs among different scenarios, and innermost ADMM-PMP, for solving the OPF for each scenario and each dispatch interval. It is to be noted, that since the algorithms are nested, that's why at the end of the inner APP iterations in Figure 7, the program-flow returns to the outermost layer of Figure 6, indicated as (**) and similarly from the end of the innermost ADMM-PMP layer of Figure 8, to the inner APP layer of Figure 7, indicated as (***). In the next section, we will present the numerical results pertaining to the LASCOPF simulations.
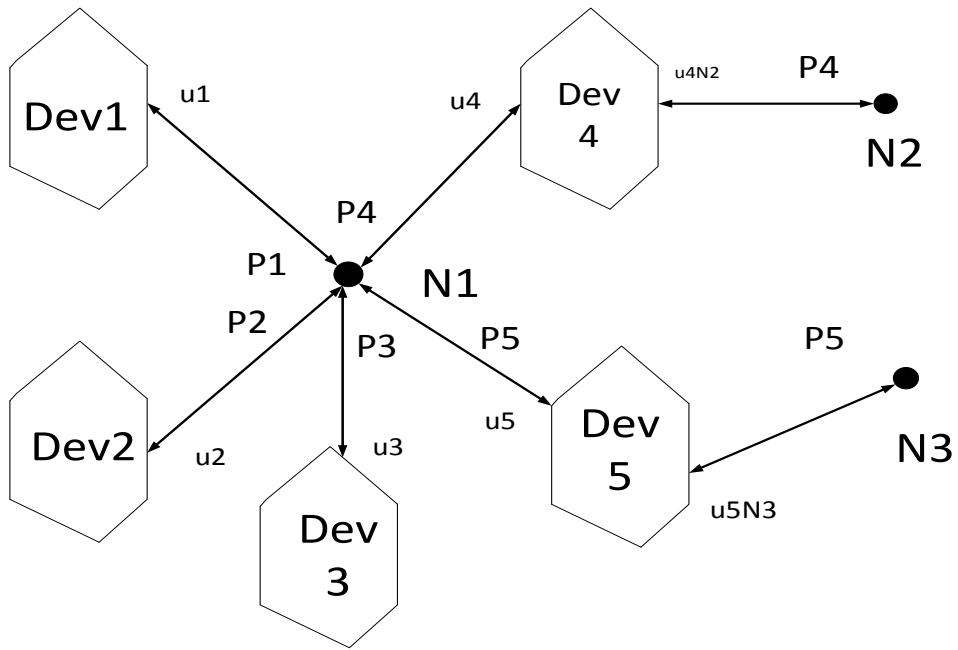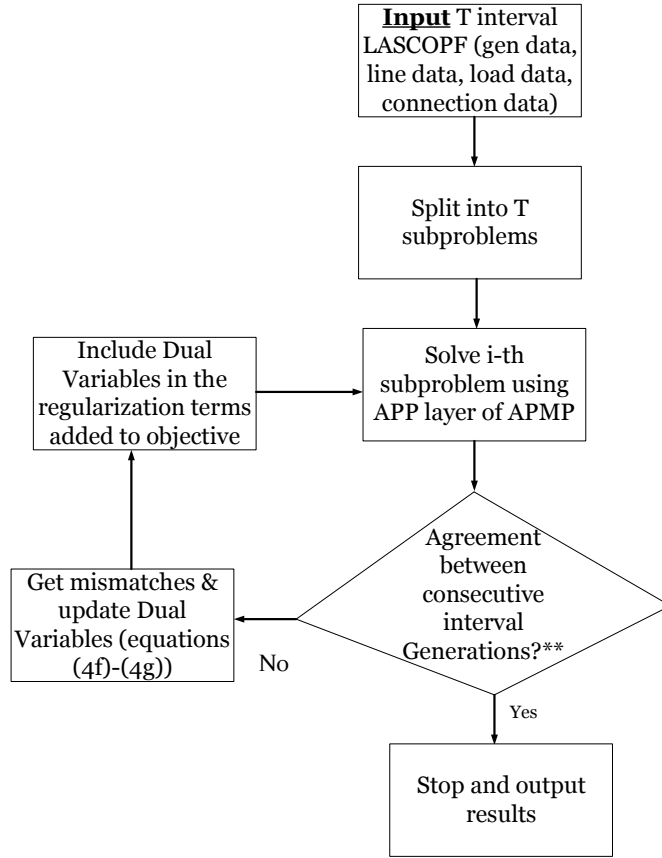
Figure 5: Generic ADMM: Broadcast and Gather

Figure 6: Flowchart for the outermost APP layer of APMP for solving LASCOPF
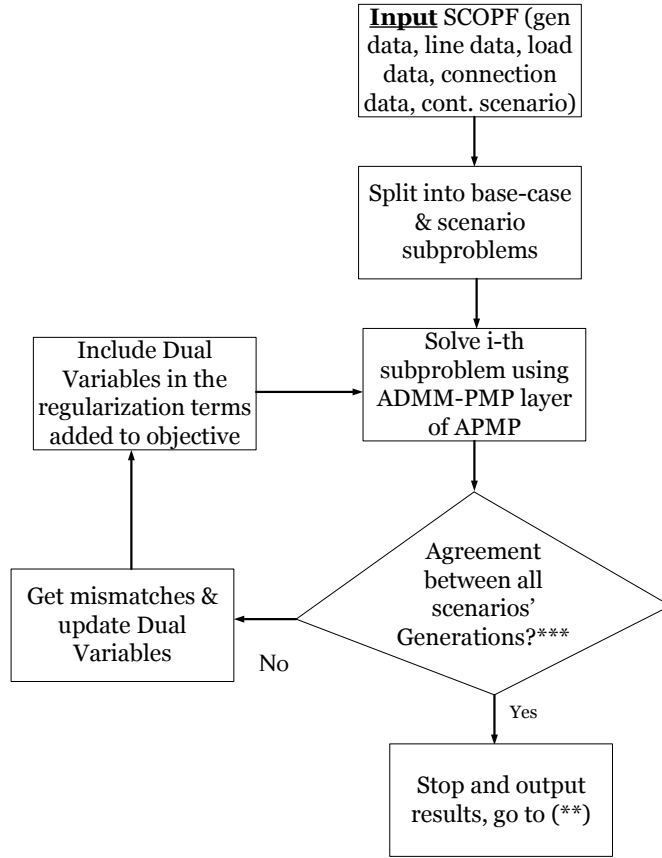
```
                    ┌─────────────────────┐
                    │ **Input** SCOPF (gen│
                    │ data, line data, load│
                    │ data, connection    │
                    │ data, cont. scenario)│
                    └─────────────────────┘
                              │
                              ▼
                    ┌─────────────────────┐
                    │ Split into base-case│
                    │    & scenario       │
                    │   subproblems       │
                    └─────────────────────┘
                              │
                              ▼
┌──────────────────┐   ┌─────────────────────┐
│ Include Dual     │   │ Solve i-th          │
│ Variables in the │──▶│ subproblem using    │
│ regularization   │   │ ADMM-PMP layer      │
│ terms added to   │   │ of APMP             │
│ objective        │   └─────────────────────┘
└──────────────────┘             │
         ▲                       ▼
         │                   ╱─────────╲
         │                  ╱ Agreement ╲
┌────────────────┐        ╱  between all  ╲
│ Get mismatches │◀──────▕   scenarios'     ▏
│ & update Dual  │        ╲ Generations?***╱
│ Variables      │   No    ╲─────────────╱
└────────────────┘              │
                             Yes│
                                ▼
                    ┌─────────────────────┐
                    │ Stop and output     │
                    │ results, go to (**) │
                    └─────────────────────┘
```

Figure 7: Flowchart for the inner APP layer for solving SCOPF

Input OPF (gen data, line data, load data, connection data)

Split into Generator, Line, Loads subproblems

Generator's prox function evaluation

Line's prox function evaluation

Load's prox function evaluation

Include Dual Variables in the regularization terms added to objective

Conn. Node (Physical Network)

Get mismatches & update Dual Variables

Is the primal residual less than tolerance?

No

Yes

Is the dual residual less than tolerance?

No

Yes

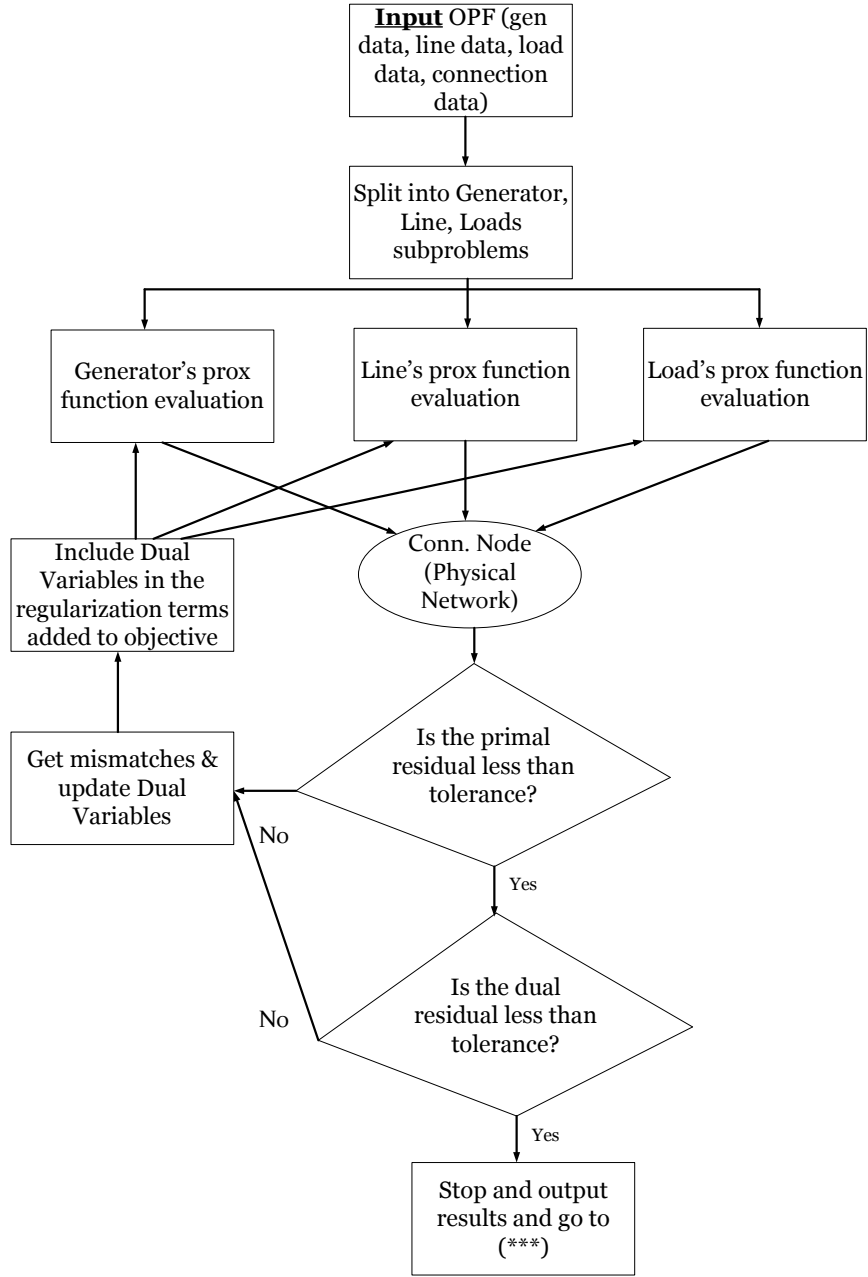Stop and output results and go to (***)

Figure 8: Flowchart for the innermost ADMM-PMP for solving OPF

7. Numerical Results

In this section, we present the results of the APMP algorithm for the LASCOPF problem instance for tracking demand variation for the IEEE Test cases [51] and for a 5 bus system, with data shown in tables 2 and 3. In table 2, A, B, and C are the quadratic, linear,

Table 2: Generator Data for 5 Bus System.

Node	A ($/MWh^2)	B($/MWh)	C($/h)	$\overline{P}$	$\underline{P}$	$\overline{R}$	$\underline{R}$	Sched. MW
1	0.0430293	20	0	332.4	0	20	-20	140.765
2	0.25	20	0	140	0	15	-15	24.2275

Table 3: Line Data for 5 Bus System.

From Node	To Node	Resistance (pu)	Reactance (pu)	Flow Limit (MW)
1	2	0.02	0.06	100
1	3	0.08	0.24	100
2	3	0.06	0.18	100
2	4	0.06	0.18	100
2	5	0.04	0.12	100
3	4	0.01	0.03	100
4	5	0.08	0.24	100

and no load coefficients of the cost curve, respectively. $\overline{P}$, $\underline{P}$, $\overline{R}$, and $\underline{R}$ are respectively, the maximum, minimum generating limits in MW and maximum and minimum ramping limits expressed in MW/dispatch interval. While performing the actual simulations, we have observed that it is very hard to solve the $(N-1)$ SCOPFs just by using ADMM-PMP, perhaps due to difficulty in choosing the appropriate tuning parameter values. So, we have used a version of APMP for solving the SCOPFs as well, as illustrated in figure 9. Here, we apply the ADMM-PMP to solve OPFs for the base case as well as each contingency scenario separately and use an outer layer of APP over all the scenarios to reach a consensus regarding the pre-contingency dispatch (the message exchanges regarding the beliefs of each scenario for the value of dispatch, to reach consensus, are shown with the green lines). We have been successful at solving the SCOPFs using this approach for the IEEE 5, 14, 30, 48, 57, 118, and 300 bus test cases with different lines marked for contingency, and also were able to identify those problem instances that are infeasible. Thereafter, for solving the LASCOPF, we can now treat the SCOPF solver as one unit, and as figure 10 shows, use the outermost APP layer to exchange messages about the beliefs of the different interval-wise generation values to reach a consensus. All the other LASCOPF problems can be solved in the exact same manner. We have run all these simulations on a Dell Inspiron 17R laptop computer powered by a 4x Intel(R) Core(TM) i7-4500U CPU running at 1.80 GHz, with a RAM of capacity
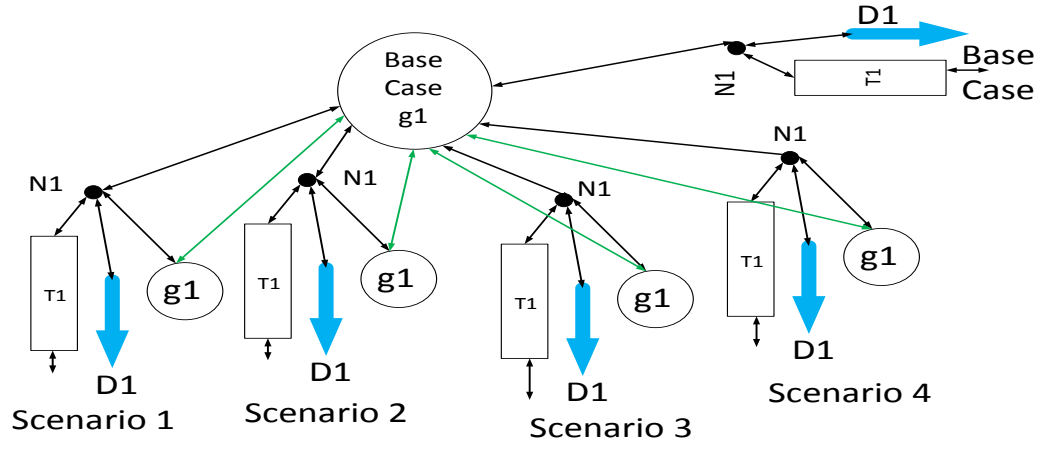
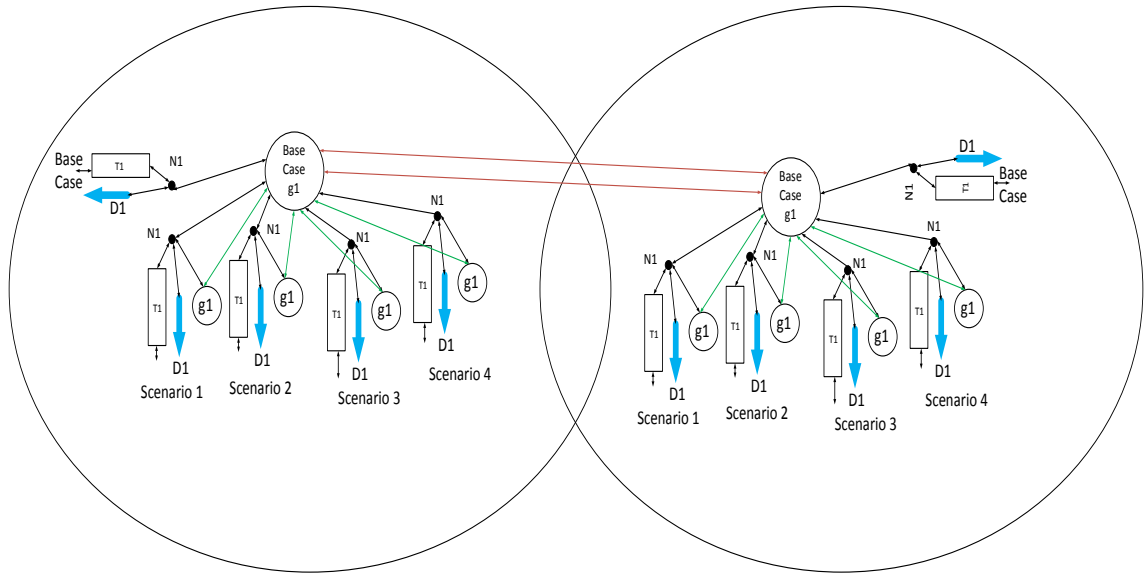Figure 9: APMP for SCOPF in each dispatch interval

Figure 10: LASCOPF for Demand Variation with APMP Schematic for Outer APP Iterations

8054 MB and the OS is Ubuntu 16.04.4 LTS. We have coded all the simulation programs in C++11, with the generator optimization solvers being implemented in two modes:

- CVXGEN custom solvers [52], [53], [54], [55] for generators' optimization for fully distributed APMP (APP+ADMM-PMP)

- GUROBI solvers for generators' optimization for fully distributed APMP (APP+ADMM-PMP)

The compiler used is GCC version 4.8.4. In the above-mentioned two methods, only the solvers are different. However, the implementation is a combination of two outer layers of coarse grained APP and one innermost layer of fine-grained ADMM-PMP. We have observed no or minimal disagreement between the results obtained by using these two solvers. However, the time taken by GUROBI is much longer than that by CVXGEN custom solvers [52], [53], [54], [55]. Hence, we will only present the results we obtained using CVXGEN. We have experimented with IEEE test cases with 5, 14, 30, 48, 57, 118, and 300 bus systems, with contingencies of some or all lines analyzed for contingencies, and for several of dispatch intervals. The simulation software that we have designed can determine (based on a combination logic of the number of iterations and convergence trend of the residuals of the different stages of the algorithm, as well as the different parts of the system) whether a particular problem instance is solvable or feasible or not. Out of those cases that we found out to be feasible, we will present here the results of the following cases:

- 5 bus system with each line analyzed for contingency

- 14 bus system with each line analyzed for contingency

- 30 bus system with lines (from node-to node) (5-7), (6-28), (10-21), and (12-16) analyzed for contingency

- 48 bus system with lines (1-5), (2-6), and (3-24) analyzed for contingency

- 57 bus system with lines (9-12), (12-17), (38-48), and (24-26) analyzed for contingency

- 118 bus system with lines (1-3) and (2-12) analyzed for contingency

- 300 bus system with lines (7-131) and (201-204) analyzed for contingency

Table 4 below lists the simulation metrics of the above-mentioned cases and the ST is the simulation time in seconds that it takes to solve the respective problems, if we implement a fully and complete nested parallelization. The numbers there represent the residual of the outermost APP layer of the algorithm, which is the norm-2 of the disagreement between the different generation MW beliefs regarding the present, previous, and next interval generation values. The ** sign next to the residuals mean that the outer iteration has already converged by the corresponding iteration count and we have fixed the subsequent residual values, just
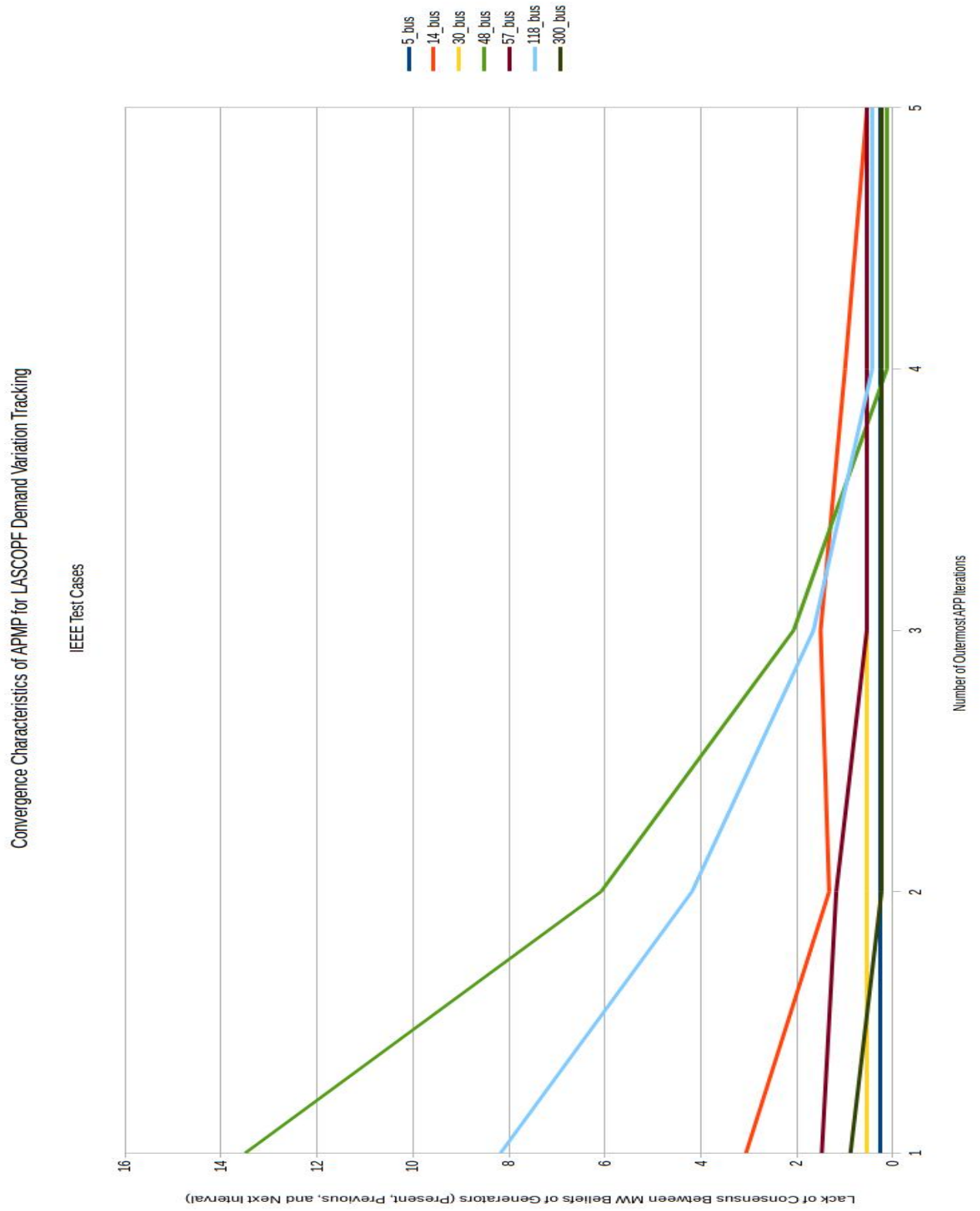
26

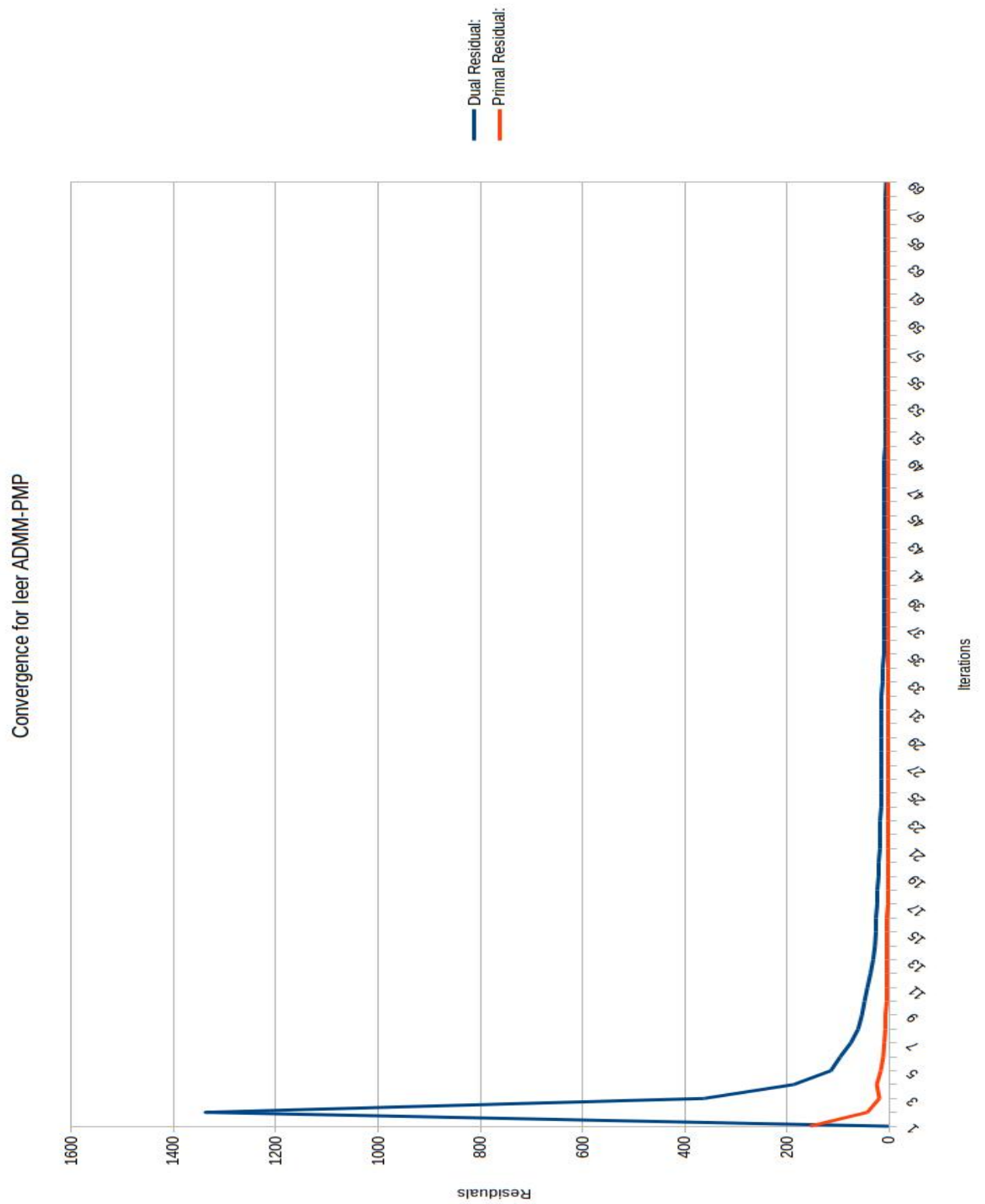Figure 11: Convergence Characteristics of IEEE Test Systems

Figure 12: Convergence Characteristics of Innermost ADMM-PMP for 300 bus case (contingency scenario:2, dispatch interval:3)

Table 4: Convergence Metrics for the IEEE Test Cases

Iter	5 Bus	14 Bus	30 Bus	48 Bus	57 Bus	118 Bus	300 Bus
1	0.2673	3.05719	0.537591	13.493	1.47779	8.17101	0.8805
2	**0.2673	1.32581	**0.537591	6.0792	1.17792	4.18356	0.2317
3	**0.2673	1.50355	**0.537591	2.06586	0.526238	1.6507	**0.2317
4	**0.2673	0.993954	**0.537591	0.111532	**0.526238	0.431	**0.2317
5	**0.2673	0.534623	**0.537591	**0.111532	**0.526238	**0.431	**0.2317
ST (s)	0.676335	205.667	1.28923	79.163	162.108	848.727	918.707

to help us draw the plot, that appears in fig 11. The load variation data for the 5 bus and 14 bus cases over 5 dispatch intervals, that we have simulated for, are explicitly shown in tables 5 and 6 respectively. The ones for the 30, 48, 57, 118, and 300 bus systems are shown as only the variations from the base case value (that appear on the UW IEEE test case archive https://labs.ece.uw.edu/pstca/) of the particular loads in particular intervals in tables 7, 8, 9, 10, and 11 respectively. The base case values occur in the first interval. For the innermost OPF simulations, which we solve by using ADMM-PMP, we have used the primal and dual residual tolerances to be 0.06 and 0.6, respectively (the reason, we use a higher value for the dual residual is that, we have observed that we get accurate enough solution for even somewhat higher value of the dual residual) and we have used a discrete version of proportional+derivative controller to adjust the value of ρ for the first 3000 iterations, such that at each iteration the relationship $\rho \times \epsilon_{primal} = \epsilon_{dual}$ is maintained, where ϵ_{primal} and ϵ_{dual} are respectively the primal and dual residuals. After the first 3000 iterations, if the algorithm hasn't still converged, then ρ is held fixed at the last value. The initial value of ρ at the beginning of the iterations is taken as 1. We have shown in Figure 12, one typical plot for the first 70 iterations of the innermost ADMM-PMP iterations for the 300 bus case for contingency scenario:2 and dispatch interval:3. This set of ADMM-PMP iterations was executed for the last iteration of the outer APP layers, both for consensus among scenarios, as well as, among dispatch intervals. The following are the APP parameters for outer iterations for attaining consensus among the different base/contingency scenarios for solving SCOPFs.

- $\alpha_{\text{SCOPF}} = 5$ for $\nu \leq 5$, $\alpha_{\text{SCOPF}} = 3$ for $5 < \nu \leq 10$, $\alpha_{\text{SCOPF}} = 2.5$ for $10 < \nu \leq 15$, $\alpha_{\text{SCOPF}} = 1.25$ for $15 < \nu \leq 20$, and $\alpha_{\text{SCOPF}} = 0.5$ for $\nu > 20$, $\beta_{\text{SCOPF}} = 200$, $\gamma_{\text{SCOPF}} = 100$ (ν is the APP iteration count)

- Final Tolerance ϵ_{SCOPF}: 0.7

Following are the APP parameters for outermost iterations for attaining consensus among the different MW outputs in different intervals, limited by ramp-rate constraints.

- $\alpha_{\text{LASCOPF}} = 10$ for $\mu_{APP} \leq 5$, $\alpha_{\text{LASCOPF}} = 5$ for $5 < \mu_{APP} \leq 10$, $\alpha_{\text{LASCOPF}} = 2.5$ for $10 < \mu_{APP} \leq 15$, $\alpha_{\text{LASCOPF}} = 1.25$ for $15 < \mu_{APP} \leq 20$, and $\alpha_{\text{LASCOPF}} = 0.5$ for $\mu_{APP} > 20$, $\beta_{\text{LASCOPF}} = 200$, $\gamma_{\text{LASCOPF}} = 100$ (μ_{APP} is the outermost APP iteration count)

- Final Tolerance $\epsilon_{\text{LASCOPF}}$: 0.6

As can be seen, for both the inner and outer APP iterations, we have used changing step-length. The way we chose those values are ad-hoc and gives a reasonable balance between accuracy and solve-time.   Figure 11 shows the outer APP convergence for this example

Table 5: Variation of Load for 5 Bus System.

Conn. Node	MW, Int.-1	MW, In.-2	MW, Int.-3	MW, Int.-4	MW, Int.-5
2	20	30	20	20	20
3	45	40	43	43	43
4	40	40	45	45	45
5	60	65	65	65	65

Table 6: Variation of Load for 14 Bus System.

Conn. Node	MW, Int.-1	MW, In.-2	MW, Int.-3	MW, Int.-4	MW, Int.-5
2	21.7	16.7	26.7	24.2	14.2
3	94.2	89.2	99.2	96.7	86.7
4	47.8	42.8	52.8	50.3	40.3
5	7.6	2.6	12.6	10.1	0.1
6	11.2	6.2	16.2	13.7	3.7
9	29.5	24.5	34.5	32	22
10	9	4	14	11.5	1.5
11	3.5	1.5	8.5	6	4
12	6.1	1.1	11.1	8.6	1.4
13	13.5	8.5	18.5	16	6
14	14.9	9.9	19.9	17.4	7.4

Table 7: Variation of Load for 30 Bus System.

Conn. Node	MW, Int.-2	MW, In.-3	MW, Int.-4	MW, Int.-5
10	0	0	10	0
12	5	0	0	0
18	0	3	0	0
24	0	0	0	12

problem. As future research, we would like to explore proven means of tuning the path length for minimum number of iterations to converge.

Table 8: Variation of Load for 48 Bus System.

Conn. Node	MW, Int.-2	MW, In.-3	MW, Int.-4	MW, Int.-5
5	0	100	0	0
7	0	0	0	100
19	0	22	0	0
20	40	0	0	0
27	0	0	0	-20
29	0	0	-47	0
32	20	0	-14	0
38	0	-100	0	0
44	0	0	100	0

Table 9: Variation of Load for 57 Bus System.

Conn. Node	MW, Int.-2	MW, In.-3	MW, Int.-4	MW, Int.-5
8	5	0	0	0
10	0	0	0	5
14	0	3	0	0
16	0	0	-20	0
25	0.4	0	0	0
29	0	0	30	0
32	0	1	0	0
41	0	3.7	0	0
47	0	6	0	0

8. Conclusion

In this paper, we have presented the Look-Ahead SCOPF (LASCOPF) considering variation of load demand over several dispatch time intervals and ability of the LASCOPF model to successfully track the temporal demand variation. We have also presented a completely decentralized (i.e. without any central coordinator) computational algorithm to implement the calculations for such a massive scale problem. Through the simulation examples, we have demonstrated the scalability of the algorithm and the effectiveness of the methodology for dispatching systems where load varies over time. The next step in the research in this direction will be to represent the post-contingency states corresponding to each contingency scenario and the system restoration to security over multiple dispatch intervals. We will be presenting that topic in our next paper.

Appendix A. Acknowledgment

Table 10: Variation of Load for 118 Bus System.

Conn. Node	MW, Int.-2	MW, In.-3	MW, Int.-4	MW, Int.-5
36	0	0	20	0
51	-20	0	0	0
70	0	-10	0	0
77	0	0	0	40

Table 11: Variation of Load for 300 Bus System.

Conn. Node	MW, Int.-2	MW, In.-3	MW, Int.-4	MW, Int.-5
1	1	0	0	0
15	0	4	0	0
145	1	0	0	0
209	0	0	12	0
217	0	0	0	20
9533	1	0	0	0

References

[1] M. Kraning, E. Chu, J. Lavaei, S. Boyd, Dynamic network energy management via proximal message passing, Foundations and Trends in Optimization 1 (2).

[2] S. Chakrabarti, M. Kraning, E. Chu, R. Baldick, S. Boyd, Security constrained optimal power flow via proximal message passing, Proceedings of the 2014 IEEE PES Clemson Power Systems Conference.

[3] K. Bai, H. Wei, Y. Wang, Semidefinite programming for optimal power flow problems, International Journal of Electric Power & Energy Systems 30 (6-7) (2008) 383–392.

[4] J. Lavaei, S. Low, Zero duality gap in optimal power flow problem, IEEE Transactions on Power Systems 27 (1) (2012) 92–107.

[5] S. Sojoudi, J. Lavaei, Physics of power networks makes hard problems easy to solve, To appear, IEEE Power & Energy Society General Meeting.

[6] S. Sojoudi, J. Lavaei, Network topologies guaranteeing zero duality gap for optimal power flow problem, To appear, IEEE Power & Energy Society General Meeting.

[7] J. Lavaei, D. Tse, B. Zhang, Geometry of power flows in tree networks, To appear, IEEE Power & Energy Society General Meeting.

[8] T. Erseghe, Distributed optimal power flow using admm, Power Systems, IEEE Transactions on PP (99) (2014) 1–11. doi:10.1109/TPWRS.2014.2306495.

[9] L. Liu, A. Khodaei, W. Yin, Z. Han, A distribute parallel approach for big data scale optimal power flow with security constraints, in: Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on, 2013, pp. 774–778. doi:10.1109/SmartGridComm.2013.6688053.

[10] S. Chakrabarti, M. R. Hesamzadeh, Financial transmission rights: A new proposal, in: 2018 15th International Conference on the European Energy Market (EEM), 2018, pp. 1–5. doi:10.1109/EEM.2018.8469941.

[11] S. J. Qin, T. A. Badgwell, A survey of industrial model predictive control technology, Control Engineering Practice 11 (7) (2003) 733–764.

[12] E. G. Cho, K. A. Thoney, T. J. Hodgson, R. E. King, Supply chain planning: Rolling horizon scheduling of multi-factory supply chains, in: Proceedings of the 35th conference on Winter simulation: driving innovation, 2003, pp. 1409–1416.

[13] F. Herzog, Strategic portfolio management for long-term investments: An optimal control approach, Ph.D. thesis, ETH, Zurich (2005).

[14] K. T. Talluri, G. J. V. Ryzin, The Theory and Practice of Revenue Management, Springer, 2004.

[15] M. Kraning, Y. Wang, E. Akuiyibo, S. Boyd, Operation and configuration of a storage portfolio via convex optimization, in: Proceedings of the 18th IFAC World Congress, 2011, pp. 10487–10492.

[16] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends in Machine Learning 3 (2011) 1–122.

[17] J. Carpentier, Contribution to the economic dispatch problem, Bull. Soc. Francaise Elect. 3 (8) (1962) 431–447.

[18] S. Chakrabarti, Study of uplan based resources planning & analysis by power generation utilities in the deregulated electricity market, Master's thesis, The University of Texas at Austin (2010).

[19] H. H. Happ, Optimal power dispatch - a comprehensive survey, IEEE Transactions on Power Apparatus and Systems 96 (3) (1977) 841–854.

[20] B. H. Chowdhury, S. Rahman, A review of recent advances in economic dispatch, IEEE Transactions on Power Systems 5 (4) (1990) 1248–1259.

[21] M. B. Cain, R. P. ONeill, A. Castillo, History of optimal power flow and formulations.

[22] M. Farivar, S. H. Low, Branch flow model: Relaxations and convexificationpart i, IEEE Transactions on Power Systems 28 (3) (2013) 2554–2564.

[23] S. H. Low, Convex relaxation of optimal power flowpart i: Formulations and equivalence, IEEE Transactions on Control of Network Systems 1 (1) (2014) 15–27.

[24] E. Dall'Anese, H. Zhu, G. B. Giannakis, Distributed optimal power flow for smart microgrids, IEEE Transactions on Smart Grid 4 (3) (2013) 1464–1475.

[25] W. Zheng, W. Wu, B. Zhang, H. Sun, Y. Liu, A fully distributed reactive power optimization and control method for active distribution networks, IEEE Transactions on Smart Grid 7 (2) (2015) 1021–1033.

[26] R. Madani, A. Kalbat, J. Lavaei, Admm for sparse semidefinite programming with applications to optimal power flow problem, 2015, pp. 5932–5939. doi:10.1109/CDC.2015.7403152.

[27] S. Mhanna, A. C. Chapman, G. Verbi, Component-based dual decomposition methods for the opf problem, Sustainable Energy, Grids and Networks 16 (2018) 91 – 110. doi:https://doi.org/10.1016/j.segan.2018.04.003.
URL http://www.sciencedirect.com/science/article/pii/S2352467718300389

[28] D. K. Molzahn, F. Drfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, J. Lavaei, A survey of distributed optimization and control algorithms for electric power systems, IEEE Transactions on Smart Grid 8 (6) (2017) 2941–2962. doi:10.1109/TSG.2017.2720471.

[29] A. Kargarian, J. Mohammadi, J. Guo, S. Chakrabarti, M. Barati, G. Hug, S. Kar, R. Baldick, Toward distributed/decentralized dc optimal power flow implementation in future electric power systems, IEEE Transactions on Smart Grid 9 (4) (2018) 2574–2594. doi:10.1109/TSG.2016.2614904.

[30] B. Stott, O. Alsac, A. Monticelli, Security analysis and optimization, Proceedings of the IEEE 75 (12) (1987) 1623–1644. doi:10.1109/PROC.1987.13931.

[31] N.-Y. Chiang, A. Grothey, Solving security constrained optimal power flow problems by a structure exploiting interior point method, Tech. Rep. Technical Report ERGO-11-014, School of Mathematics, The University of Edinburgh (jun 2011).

[32] D. Phan, J. Kalagnanam, Distributed methods for solving the security-constrained optimal power flow problem, Proceedings of the 2012 IEEE PES Innovative Smart Grid Technologies Conference.

[33] R. Glowinski, A. Marrocco, Sur l'approximation, par elements finis d'ordre un, et la resolution, par penalisation-dualité, d'une classe de problems de Dirichlet non lineares, Revue Française d'Automatique, Informatique, et Recherche Opérationelle 9 (1975) 41–76.

[34] D. Gabay, Applications of the method of multipliers to variational inequalities, in: M. Fortin, R. Glowinski (Eds.), Augmented Lagrangian Methods: Applications to the Solution of Boundary-Value Problems, North-Holland: Amsterdam, 1983.

[35] M. Fortin, R. Glowinski, Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems, North-Holland: Amsterdam, 1983.

[36] J. Eckstein, M. Fukushima, Some reformulations and applications of the alternating direction method of multipliers, Large Scale Optimization: State of the Art (1993) 119–138.

[37] J. Eckstein, Parallel alternating direction multiplier decomposition of convex programs, Journal of Optimization Theory and Applications 80 (1) (1994) 39–62.

[38] B. H. Kim, R. Baldick, Coarse-grained distributed optimal power flow, IEEE Transactions on Power Systems 12 (2) (1997) 932–939.

[39] B. H. Kim, R. Baldick, A comparison of distributed optimal power flow algorithms, IEEE Transactions on Power Systems 15 (2) (2000) 599–604.

[40] F. Capitanescu, J. M. Ramos, P. Panciatici, D. Kirschen, A. M. Marcolini, L. Platbrood, L. Wehenkel, State-of-the-art, challenges, and future trends in security constrained optimal power flow, Electric Power Systems Research 81 (8) (2011) 1731–1741.

[41] M. Lubin, Y. Dvorkin, S. Backhaus, A robust approach to chance constrained optimal power flow with renewable generation, IEEE Transactions on Power Systems 31 (5) (2015) 3840–3849.

[42] N. Amjady, H. Sharifzadeh, Security constrained optimal power flow considering detailed generator model by a new robust differential evolution algorithm, Electric Power Systems Research 81 (2) (2011) 740–749.

[43] M. La Scala, M. Trovato, C. Antonelli, On-line dynamic preventive control: an algorithm for transient security dispatch, IEEE Transactions on Power Systems 13 (2) (1998) 601–610.

[44] A. J. Monticelli, M. Pereira, S. Granville, Security constrained optimal power flow with post-continaency corrective scheduling, Power Systems, IEEE Transactions on 2 (1987) 175 – 180. doi:10.1109/TPWRS.1987.4335095.

[45] G. Cohen, Optimization by decomposition and coordination: A unified approach, Automatic Control, IEEE Transactions on 23 (2) (1978) 222–232. doi:10.1109/TAC.1978.1101718.

[46] G. Cohen, Auxiliary problem principle and decomposition of optimization problems, Journal of Optimization Theory and Applications 32 (3) (1980) 277–305. doi:10.1007/BF00934554.
URL http://dx.doi.org/10.1007/BF00934554

[47] R. Baldick, B. Kim, C. Chase, Y. Lao, A fast distributed implementation of optimal power flow, IEEE Transactions on Power Systems 31 (3).

[48] R. Ebrahimian, R. Baldick, State estimation distributed processing [for power systems], Power Systems, IEEE Transactions on 15 (4) (2000) 1240–1246. doi:10.1109/59.898096.

[49] J. Batut, A. Renaud, Daily generation scheduling optimization with transmission constraints: a new class of algorithms, Power Systems, IEEE Transactions on 7 (3) (1992) 982–989. doi:10.1109/59.207311.

[50] S. Chakrabarti, Post-contingency states representation & redispatch for restoration in power systems operation, Ph.D. thesis, The University of Texas at Austin (2017).

[51] U. of Washington, Power systems test case archive.
URL https://labs.ece.uw.edu/pstca/

[52] J. Mattingley, S. Boyd, Cvxgen: a code generator for embedded convex optimization, Springer Optim. Eng.

[53] Y. W. Jacob Mattingley, S. Boyd, Code generation for receding horizon control, Stanford University.

[54] J. Mattingley, S. Boyd, Real time convex optimization in signal processing, IEEE Signal Processing Magazine.

[55] J. Mattingley, S. Boyd, Automatic code generation for real-time convex optimization, Convex Optimization in Signal Processing and Communications.