Clustering-based Decentralized Optimization Approaches for DC Optimal Power Flow

Kai Zhang*, Sarmad Hanif†, Dante Recalde‡
TUM CREATE Limited,
#10-02 CREATE Tower, Singapore 138602
Email: *kai.zhang@tum-create.edu.sg, †sarmad.hanif@tum-create.edu.sg, ‡dante.melo@tum-create.edu.sg

***Abstract*—This paper studies two decentralized scheme to solve DC optimal power flow (DC-OPF). The first scheme considers the decomposition of DC-OPF based upon augmented Lagrangian relaxation and uses alternating direction method of multipliers (ADMM) algorithm to solve the consensus optimization problem. An adaptive penalty method is proposed for the ADMM algorithm to improve the convergence performance. The second scheme utilizes Karush-Kuhn-Tucker (KKT) conditions and solves the coupled linear equations of DC-OPF directly. We show the impact of different cluster formations on both schemes. Both schemes are evaluated in terms of flexibility, robustness and iteration time using the IEEE 14-bus test system.**

***Index Terms*—Decentralized control, DC-OPF, ADMM, KKT, Clustering**

I. Introduction

Traditionally, power systems have been operated as clusters due to historical and geopolitical reasons [1], [2]. The high number of aggregators and other market participants present in smart grids increase the complexity of the system. This may create bottlenecks in the communication infrastructure if the optimal power flow (OPF) problem is solved in a centralized manner [3]. The high number of controllable nodes makes the DC-OPF very challenging to solve. Hence, clustering can be used to reduce the computational burden as well as the complexity of the central information collection and processing. In the context of deregulation carried out in power systems, the market requires each regional operator to solve the optimal power flow problem in a cooperative way while protecting their private information. The authors in [4] propose a hierarchical clustering formation method based on Jacobian matrix from power flow solutions.

Many algorithms to solve the OPF in a decentralized manner have been proposed in the literature [5]–[8]. The authors in [8] compare existing algorithms used to solve DC-OPF in a decentralized/distributed way. These algorithms can be divided into two main categories: decomposition of optimization problems based on augmented Lagrangian relaxation, and decomposition based on KKT conditions. In respect of the implementation of the decentralized approaches, methods from both categories present pro and cons in terms of flexibility, robustness, communication requirement and performance. However, these practical issues have not yet been studied extensively in the literature. Furthermore, these studies do not include the communications in the evaluation of the algorithms.

978-1-5386-4950-3/17/$31.0

In this paper ADMM and KKT-based algorithm are investigated to study the unique requirement of solving DC-OPF for cluster-based control. A fast ADMM approach using adaptive penalty factor is proposed. This method is compared against a KKT-based algorithm using different cluster structures. This paper explores the impact of cluster formation on the number of iterations required for convergence and the computation time required to solve/compute the solution at each iteration. The performance of the proposed method is evaluated using the IEEE 14-bus test case.

The paper is arranged as follows. In Section II, the problem formulation of the DC-OPF is reviewed and the comparison of two decentralized optimization strategies are given with respect to the algorithm architecture. Section III introduces the framework employed to solve the cluster-based DC-OPF using the ADMM algorithm and the KKT-condition based method. In Section IV, numerical experiment results are presented for the performance of the given methods. The conclusion is given in Section V.

II. Decentralized Architectures for Solving DC-OPF

For the transmission system, it can be assumed that the R/X ratio is very small along with voltage magnitude being close to one per unit as well as the voltage angle difference very small [9]. This allows us to write the DC-OPF problem as follows:

$$\min \quad \sum_{i\in\Omega_B} \left[c_2 P_{Gi}^2 + c_1 P_{Gi} + c_0\right] \tag{1a}$$

s.t.

$$\sum_{j\in\Omega_i} b_{ij}(\theta_i - \theta_j) = P_{Gi} - P_{Li} \qquad \forall i \in \Omega_B \tag{1b}$$

$$\underline{P}_{Gi} \le P_{Gi} \le \overline{P}_{Gi} \qquad \forall i \in \Omega_G \tag{1c}$$

$$|b_{ij}(\theta_i - \theta_j)| \le \overline{P}_{ij} \qquad \forall ij \in \Omega_L \tag{1d}$$

where, (1a) depict the total cost for the generators, (1b) ensures the power balance for all nodes, (1c) limits the generators output and (1d) depict the branch flow relation for all lines. A summary of the optimization variables is given in Table I.

A. ADMM-based Decentralized Architecture

In [10], an ADMM-based approach is used to solve the consensus optimization problem. For this paper, we adopt a similar idea for solving the DC-OPF. The basic idea of this

Table I
VARIABLES FOR DC-OPF AND CLUSTERING

Symbol	Description
P_{Gi}	generations at bus i
P_{Li}	load at bus i
c_0, c_1, c_2	coefficients of n-th order polynomial cost
θ_i	voltage angle at bus i
Ω_i	set of all the buses connected to bus i
Ω_L	set of all lines in the network
Ω_B	set of all buses in the network
Ω_G	set of generations
b_{ij}	line susceptance from bus i to bus j
$\underline{P}_{Gi}$,$\overline{P}_{Gi}$	generation lower and upper bound at bus i
$\overline{P}_{ij}$	line capacity from bus i to bus j
Ω_C	set of clusters $\Omega_C = \{1, 2, ..., M\}$
Λ_m	set of buses in cluster m : $\Lambda_m \subseteq \Omega_B$
Φ_m	set of buses in neighboring cluster that connect to cluster m
Ψ_m	set of buses in cluster m that connect to neighboring clusters
Π_m	set of coupled buses for cluster m: $\Pi_m = \Phi_m \cup \Phi_m$
Π	set of coupled buses in the network $\Pi = \sum_1^M \Pi_m$

approach is to decompose the original problems into separable parts. To demonstrate it, consider a generic objective function split in N independent parts:

$$\min \quad \sum_{i=1}^{N} f_i(\boldsymbol{x_i}) \tag{2}$$

where the local variables $\boldsymbol{x_i} \in \mathbb{R}^{n_i}$ also contains selected number of global variables $\boldsymbol{z} \in \mathbb{R}^n$. Figure 1 gives an example of the projection map between the local and global variables for the case of $N = 3$. Let $\boldsymbol{z_i} \in \mathbb{R}^{n_i}$ denote the corresponding global variable collection for $\boldsymbol{x_i}$. The correspondence of local variable $\boldsymbol{x_i}$ to the global variable z_g is described using a projection map $g = \mathcal{G}(i, j)$, where $(\boldsymbol{x_i})_j = z_g$.

By using the definition in (2) and the projection map procedure shown in Fig. 1, we can now formulate the DC-OPF problem in (1) as the following consensus optimization problem:

$$\min \quad \sum_{i=1}^{N} f_i(\boldsymbol{x_i}) \tag{3a}$$

s.t.

$$\boldsymbol{x_i} - \widetilde{\boldsymbol{z}}_{\boldsymbol{i}} = \boldsymbol{0}, \quad \forall i = 1, ..., N \tag{3b}$$

for each subproblem i, the variable $\widetilde{\boldsymbol{z}}_{\boldsymbol{i}} \in \mathbb{R}^{n_i}$ contains the corresponding value that the local variable $\boldsymbol{x_i}$ should be. The goal of the consensus optimization problem is to solve it in such a way that each subproblem can be solved by its own computation unit. As shown in Section III-A, the consensus problem in (3) is solved using an ADMM-based iterative algorithm [10].

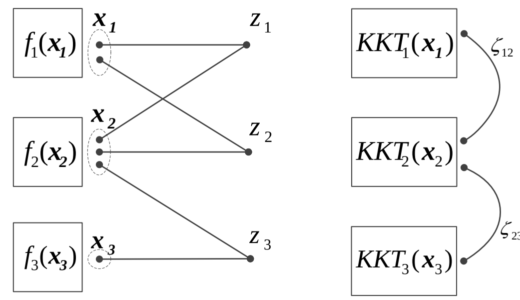

Fig. 1. Optimization problem decomposition for ADMM (right) and KKT-based (left) methods. KKT_i denotes i^{th} subproblem of KKT conditions shown in (5). Moreover, KKT_i has mapping functions ζ_{ij} representing the connection between the variable $\boldsymbol{x_i}$ for KKT_i and the variable $\boldsymbol{x_j}$ for the connected KKT_j.

B. KKT-based Decentralized Architecture

The KKT-based decentralized architecture exploits the actual physical connections (nodal voltage angles) between individual and neighboring buses [11]. Consider the following generic optimization problem:

$$\min \quad \sum_{i=1}^{N} f_i(\boldsymbol{x_i}) \tag{4a}$$

s.t.

$$h(\boldsymbol{x_i}) \leq \boldsymbol{0}, \tag{4b}$$

$$g(\boldsymbol{x_i}) = \boldsymbol{0}, \quad \forall i = 1, ..., N. \tag{4c}$$

The KKT conditions for the optimization problem are:

$$\partial f_i(\boldsymbol{x_i}) + \nu_i \cdot \partial h(\boldsymbol{x_i}) + \mu_i \cdot \partial g(\boldsymbol{x_i}) = 0 \tag{5a}$$

$$\nu_i h(\boldsymbol{x_i}) = 0, \tag{5b}$$

$$\nu_i \geq 0, \tag{5c}$$

$$\text{(4b)}, \quad \text{(4c)}, \quad \forall i = 1, ..., N, \tag{5d}$$

where ν_i and μ_i are the Lagrange multipliers associated with constraints (4b) and (4c) respectively. Notice that in (5), the optimization problem of (4) has been replaced by a system of coupled linear equations. These system of equations establish a natural interconnection of neighboring nodes and their associated Lagrange multipliers. Hence, a decentralized scheme purely relying on the information flow of physically connected neighbors can be established [11]. The interconnection of neighboring nodes is reflected in the mapping function ζ_{ij} shown in Fig. 1. In Section III-B, we show an iterative algorithm for solving KKT-conditions of (5) for the application of the DC-OPF.

III. DECENTRALIZED ALGORITHMS FOR SOLVING DC-OPF

A. Fast ADMM-based algorithm

The clustering variables to be used for designing the ADMM-based decentralized schemes are defined in Section II. For each cluster, the bus angles between local and neighboring clusters are used as global variables for solving the DC-OPF using the ADMM algorithm. The set containing all such angles is given by Π_m. The cardinality of set Π is denoted by N_b. Let $\boldsymbol{\theta}^{\Pi_m} \in \mathbb{R}^{n_m}$ represents the local copy for the voltage angle vector of coupled buses for cluster m. Similarly, $\boldsymbol{\theta}^{\Pi_m+} \in \mathbb{R}^{n_m}$ represents the global variable for the voltage angle vector of coupled buses for cluster m. Furthermore, the vector of buses in set Π is represented as $\boldsymbol{\theta}^{\Pi+} \in \mathbb{R}^{N_b}$. The correspondence between the local copy and the global variable is described using the graph $g = \mathcal{G}(m,n)$. Where $(\boldsymbol{\theta}^{\Pi_m})_n = (\boldsymbol{\theta}^{\Pi+})_g$. Since the object function and the constraints are separable and can be associated to each cluster. For each cluster the following optimization problem is solved:

$$\min \sum_{i\in\Lambda_m} \left[c_2 P_{Gi}^2 + c_1 P_{Gi} + c_0\right] \tag{6a}$$

s.t.

$$\sum_{j\in\Omega_i} b_{ij}(\theta_i - \theta_j) = P_{Gi} - P_{Li} \qquad \forall i \in \Lambda_m \tag{6b}$$

$$\boldsymbol{\theta}^{\Pi_m} = \boldsymbol{\theta}^{\Pi_m+} \qquad \forall m \in \Omega_C \tag{6c}$$

$$\underline{P}_{Gi} \le P_{Gi} \le \overline{P}_{Gi} \qquad \forall i \in \Lambda_m \tag{6d}$$

$$|b_{ij}(\theta_i - \theta_j)| \le \overline{P}_{ij} \quad \forall i \in \Lambda_m, j \in \Omega_i \tag{6e}$$

Constraint (6c) ensures the copy of local angles is equal to the global variable. Let $\boldsymbol{\lambda}^{\Pi_m}, \boldsymbol{\rho}^{\Pi_m} \in \mathbb{R}^{n_m}$ denote the Lagrangian multiplier and the adaptive penalty factor respectively. The partial augmented Lagrangian with adaptive penalty factor for each cluster is given as follows:

$$\begin{aligned}\mathcal{L}_m = \sum_{i\in\Lambda_m} & \left[c_2 P_{Gi}^2 + c_1 P_{Gi} + c_0\right] \\ & + (\boldsymbol{\lambda}^{\Pi_m})^T(\boldsymbol{\theta}^{\Pi_m} - \boldsymbol{\theta}^{\Pi_m+}) \\ & \quad + \frac{1}{2}(\boldsymbol{\rho}^{\Pi_m})^T(\boldsymbol{\theta}^{\Pi_m} - \boldsymbol{\theta}^{\Pi_m+})^2\end{aligned} \tag{7}$$

The ADMM algorithm with adaptive penalty factor for solving DC-OPF is solved iteratively. For cluster m at the $(k+1)$-th step, variables are updated in the following manner:

1) **Local minimization**

$$\boldsymbol{\theta}^{\Pi_m}(k+1) := \arg \min_{\theta_i,\theta_j,\boldsymbol{\theta}^{\Pi_m},P_{Gi}} \mathcal{L}_i \tag{8}$$

$$s.t. (6b), (6d), (6e)$$

2) **Global variable update**

$$(\boldsymbol{\theta}^{\Pi+})_g(k+1) = \frac{1}{N_g} \sum_{\mathcal{G}(m,n)=g} (\boldsymbol{\theta}^{\Pi_m})_n \quad \forall g \in \Psi_m \tag{9}$$

where N_g is the number of local variables that correspond to the global variable $(\boldsymbol{\theta}^{\Pi+})_g$. Each cluster updates the global variables representing the voltage angles included in the local cluster.

3) **Primal and dual residual update**

For each cluster, the primal residual $\boldsymbol{r}^{\Pi_m} \in \mathbb{R}^{n_m}$ and dual residual $\boldsymbol{s}^{\Pi_m} \in \mathbb{R}^{n_m}$ are updated as:

$$\begin{aligned}\|(\boldsymbol{r}^{\Pi_m})_n(k+1)\|_2^2 = & \left[(\boldsymbol{\theta}^{\Pi_m})_n(k+1)\right. \\ & \left. - (\boldsymbol{\theta}^{\Pi_m+})_{\mathcal{G}(m,n)}(k+1)\right]^2\end{aligned} \tag{10a}$$

$$\begin{aligned}\|(\boldsymbol{s}^{\Pi_m})_n(k+1)\|_2^2 = & \left[(\boldsymbol{\rho}^{\Pi_m})_n(k+1)\right]^2 \cdot \\ & \left[(\boldsymbol{\theta}^{\Pi+})_{\mathcal{G}(m,n)}(k+1) - (\boldsymbol{\theta}^{\Pi+})_{\mathcal{G}(m,n)}(k)\right]^2\end{aligned} \tag{10b}$$

4) **Penalty factor update**

$$\left(\boldsymbol{\rho}^{\Pi_m}\right)_n(k+1) = \begin{cases} \left(\boldsymbol{\rho}^{\Pi_m}\right)_n(k) \cdot (1+\tau^t) \\ \quad \text{if } \|\left(\boldsymbol{r}^{\Pi_m}\right)_n(k+1)\|_2^2 > \|\left(\boldsymbol{s}^{\Pi_m}\right)_n(k+1)\|_2^2 \\ \left(\boldsymbol{\rho}^{\Pi_m}\right)_n(k) \cdot (1+\tau^t)^{-1} \\ \quad \text{if } \|\left(\boldsymbol{s}^{\Pi_m}\right)_n(k+1)\|_2^2 > \|\left(\boldsymbol{r}^{\Pi_m}\right)_n(k+1)\|_2^2 \\ \left(\boldsymbol{\rho}^{\Pi_m}\right)_n(k) \\ \quad \text{otherwise} \end{cases} \tag{11}$$

where ρ is the adaptive penalty factor.

5) **Lagrangian multiplier update step**

$$\begin{aligned}\boldsymbol{\lambda}^{\Pi_m}(k+1) = & \boldsymbol{\lambda}^{\Pi_m}(k) \\ & + \boldsymbol{\rho}^{\Pi_m} \circ \left[\boldsymbol{\theta}^{\Pi_m}(k+1) - \boldsymbol{\theta}^{\Pi_m+}(k+1)\right]\end{aligned} \tag{12}$$

The operator $\circ$ denote the entry-wise multiplication of the elements.

The convergence of generic ADMM aglorithm can be defined based on the residual, objective and dual variable convergence [10]. The ADMM method introduced in [10] solves the generic consensus optimization problem with a fixed penalty factor. We adapt the definitions of local residuals in [12] to the penalty factor varying strategy in [13] to improve the convergence performance for the ADMM algorithm. The proposed method changes the penalty factor based on the relative magnitude of the primal and dual residuals. This reduces the dependency on the initial parameter value.

B. *KKT-based algorithm*

The KKT-based fully decentralized scheme is presented in this section to solve the DC-OPF problem. The general form of KKT conditions required to obtain consensus between the coupled variable is shown in (5). We deploy a similar approach in the following by defining the KKT conditions of equations (1a)-(1d) as:

$$2c_2 P_{Gi} + c_1 - \lambda_i + \overline{\mu}_i - \underline{\mu}_i = 0 \quad \forall i \in \Omega_G \tag{13a}$$

$$\begin{aligned}\lambda_i^K \cdot \sum_{j\in\Omega_i} b_{ij} - & \sum_{j\in\Omega_i} \lambda_j^K b_{ij} \\ & + \sum_{j\in\Omega B} (\mu_{ij} - \mu_{ji}) b_{ij} = 0 \quad \forall i \in \Omega_i\end{aligned} \tag{13b}$$

$$\theta_1 = 0 \tag{13c}$$

$$(1b), (1c), (1d) \tag{13d}$$

along with the complementary slackness and positive constraints on the Lagrange multipliers of constraints of (1c)-(1d). Variable λ_i^K is the Lagrange multiplier for the power balance constraint in (1b). The Lagrange multipliers $\overline{\mu}_i$ and $\underline{\mu}_i$ are associated with the maximum and minimum power generation limitations in (1c), whereas μ_{ji} represents the power flow limitations in (1d). With the assumption of convexity and strong duality, the above mentioned optimality conditions are necessary and sufficient. Therefore, solving the above system of KKT equations should also recover the optimal solution.

With respect to the update steps defined in Section III-A, the KKT-based decentralization scheme solves the system of equations in (13) as:

1) **Primal updates**

Generation set points: The sensitivity of the generation set points with respect to the KKT conditions is represented by (13a). Moreover the limits on these set points are enforced through (13d). Hence, for the given value of $\lambda_i^K(k)$ at iteration step k, these two equations can be solved using the following iterative procedure:

$$P_{Gi}(k+1) = \Big[\frac{\lambda_i^K(k) - c_1}{c_2}\Big]_{\underline{P}_{Gi}}^{\overline{P}_{Gi}}, \tag{14}$$

where k is the iteration step. Note that the operator $[.]_{\underline{P}_{Gi}}^{\overline{P}_{Gi}}$ projects P_{Gi} to its natural limits of (1c).

Nodal angles: In a similar manner, θ_i can be solved iteratively by utilizing its concerned KKT equation, i.e., (1b).

$$\theta_i(k+1) = \theta_i(k) + \alpha\Big(P_{Gi}(k) - P_{Li} - \sum_{j\in\Omega_B} b_{ij}[\theta_i(k) - \theta_j(k)]\Big) \tag{15}$$

where α is the tuning parameter.

2) **Dual updates**

Power balance dual: Note that in (14) and (15), the primal variables updates are connected to their respective dual variables. In a similar fashion to the primal update procedure, we can also update the dual variables. For each iteration step k, $\lambda_i^K(k+1)$ can be found using:

$$\begin{aligned}\lambda_i^K(k+1) &= \lambda_i^K(k) - \beta\Big(\lambda_i^K(k).\sum_{j\in\Omega_i} b_{ij} \\ &- \sum_{j\in\Omega_i}\lambda_j^K(k)b_{ij} + \sum_{j\in\Omega_B}[\mu_{ij}(k) - \mu_{ji}(k)]b_{ij}\Big) \\ &+ \gamma\Big(P_{Gi}(k) - P_{Li} - \sum_{j\in\Omega_B} b_{ij}[\theta_i(k) - \theta_j(k)]\Big)\end{aligned} \tag{16}$$

where β and γ are tuning parameters. Note that update step for λ_i^K contains two terms. Using (13b), the first term in (16) reflects the agreement of λ_i^K with its neighbouring Lagrange multiplier, i.e, λ_j^K. The second term reflects the effect of the power balance at each node (1d).

Line flow dual: In the first term of (16), the dual variable $\mu_{ij}(k)$ which shows line limit violations are updated using:

$$\mu_{ji}^{k+1} = \left[\mu_{ji}(k) - \delta\Big(\overline{P}_{ij} - b_{ij}[\theta_i(k) - \theta_j(k)]\Big)\right]^+ \tag{17}$$

where δ is a constant tuning parameter and $[.]^+$ enforces non-negativity constraint on the dual variable.

The KKT-based algorithm exploits the *consensus+innovations* type structure [11]. The tunning parameters in the consensus algorithm can be modified in a way to minimize the convergence time (see e.g. [14]). However, it requires the knowledge of the full graph structure, which is not commonly assumed in the literature [9].

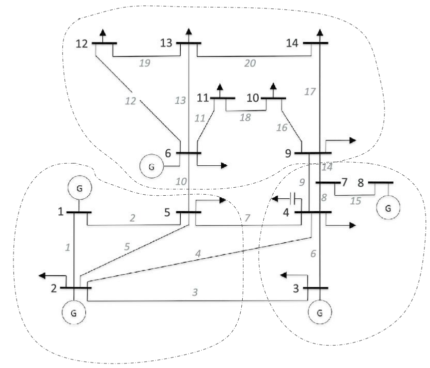

Fig. 2. IEEE 14 bus test case with three clusters

IV. Comparison of ADMM and KKT algorithm in cluster-based control

A. Architectural and Organizational Comparison

Both ADMM-based and KKT-based algorithms possess the flexibility of formating the clusters, however the impact of the cluster-based control on the algorithms is different. Notice that in comparison to the consensus problem solved in (3a) and (3b) by ADMM algorithm, KKT-based decentralized scheme exploits the information flow from the physically connected neighbors, avoiding creating local copies of global connected variables (see Fig. 1). Hence, the flexibility from ADMM-based architecture can be utilized to reorganize the DC-OPF to define each local subproblem, whereas KKT-based scheme is only solvable on a nodal basis. Therefore, a new cluster formation for KKT-based algorithm is merely a question of reorganizing the communication network. On the other side, due to the existence of local copies of the global variables in the individual subproblems, the ADMM-based algorithm is much more robust for solving the optimization problem. In terms of communication network, both algorithms require a centralized topology within the cluster. A decentralized topology exists for communication between different clusters. The centralized communication can be eliminated if each cluster is associated with only one bus. In this way, the ADMM-based algorithm can serve as a variant of the KKT-based decentralized scheme. Although this might also eradicate the inherent flexibility of the ADMM-based method in organizing flexible clusters for the whole problem.

B. Numerical Experiment on IEEE 14-Node Network

The clustering formation has a direct impact on the convergence time for ADMM algorithm, which does not apply for KKT-based method. The cluster formations is reflected in two aspects: the size and the connectivity between clusters. The connectivity between clusters can be measured by the number of coupled buses defined as the cardinality of Π_m in Section II. For a fixed network, the increment on the number of clusters means smaller cluster sizes and a higher decentralization grade for solving the DC-OPF.

Table II
CLUSTERING PARAMETERS FOR TEST CASES

Test Case	Number of Clusters	Members
Centralized	1	OPF solved by one entity
Case 1	2	c1 = {1 2 3 4 5 7 8}; c2 = {6 9 10 11 12 13 14};
Case 2	4	c1 = {1 2 5}; c2 = {3 4 7 8}; c3 = {9 10 14}; c4 = {6 11 12 13};
Case 3	7	c1 = {1 2}; c2 = {3 4}; c3 = {5 6}; c4 = {7 8 9}; c5 = {10 11}; c6 = {14}; c7 = {12 13};
Case 4	14	OPF solved by each node
Case 5	2	c1 = {1 2 5 6 11 12 13}; c2 = {3 4 7 8 9 10 14};

A case study is developed for testing the effect of different clustering formation for the IEEE 14-bus system. The clustering formation summary is shown in Table II. Case 1-4 are developed to compare the influence of different cluster sizes and Case 5 is designed to study the impact of the connectivity. Note that for Case 5 the clusters are linked through 8 coupled buses whereas it only has 5 coupled buses in Case 1.

We adopt two criteria for the performance evaluation. The standard deviation for the power generations values of the given algorithm and the centralized solution is given as:

$$P_{dev} = \sqrt{\frac{1}{M}\sum_{1}^{M}(P_{Gi} - P_{Gi}^{opt})^2} \quad (18)$$

The second criterion evaluates the total cost function deviation of the given algorithm and the optimal value calculated by the centralized solution.

$$rel = \frac{|f_{opt} - f_{algorithm}|}{f_{opt}} \quad (19)$$

Fig. 3 and 4 illustrate the simulation results. It is worth observing that each ADMM algorithm iteration require two times information update between neighbors while only one information update is needed for the KKT-based method. Hence, the x-axis is assigned with communication iterations. One iteration for ADMM algorithm means two communication iterations while the number for the KKT-based algorithm is one. The results in both figures make intuitive sense, more clusters means more iterations for ADMM algorithm to converge to the optimum. On the other hand, by comparison of test cases 1 and 5, more coupled buses between clusters will also accelerate the convergence but requires a larger amount of the information exchange. For KKT-condition based method, the clustering formation does not affect the convergence time since the KKT-based algorithm is decomposed on a nodal basis. An overview of the test results is given in Table III.

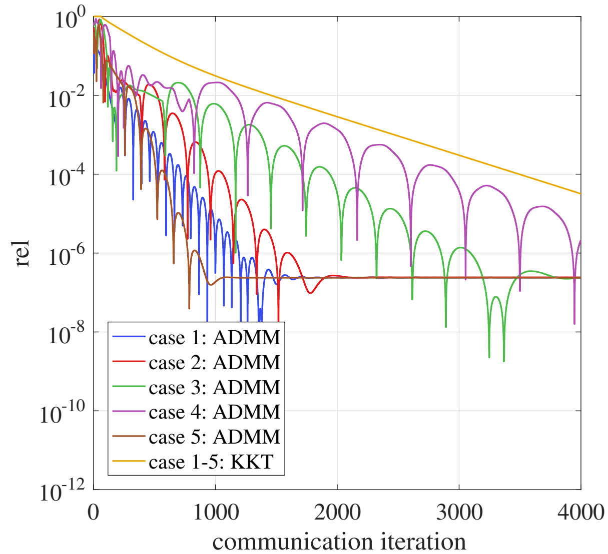

Fig. 3. Deviation to optimal generation cost

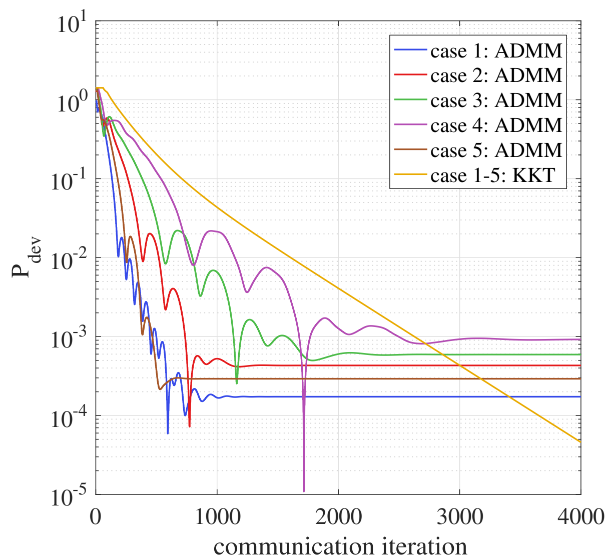

Fig. 4. Standard deviation to optimal power generation

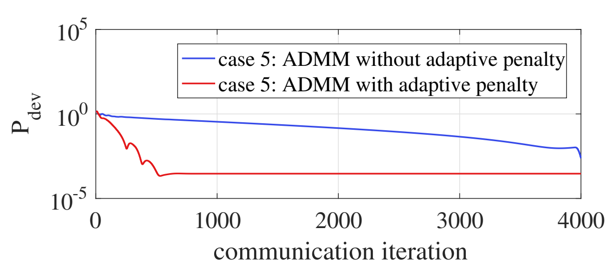

Fig. 5. Adaptive penalty factor comparison

Table III
COMPARISON OF DECENTRALIZED ALGORITHMS

	ADMM Algorithm					KKT-based Algorithm				
Shared Variables	1) θ: local optimal angles 2) θ: global average angles					1) θ_i: nodal angles 2) P_{Gi} :nodal generations 3) μ_{ij},λ_i^K: Lagrangian multiplier				
Required iterations	Case 1	Case 2	Case 3	Case 4	Case 5	Case 1	Case 2	Case 3	Case 4	Case5
	380	550	990	1485	260	3490				
Computation time/iteration	0.33 - 0.57 s					0.12 - 0.23 ms				
information updates/iteration	2					1				

Fig. 5 illustrates the influence of the adaptive penalty to the convergence speed of the ADMM-based algorithm.

In practice, both methods require the two-way communication between clusters. The communication requirement for ADMM algorithm depends on the cluster-formation. The maximal number of coupled buses between clusters in the network decides the minimal communication bandwidth for the implementation. Although ADMM algorithm requires larger computational time, it shows robustness against communication package loss and failure. The KKT-based method only needs to solve several linear equations and therefore the computation time is almost neglectable in comparison with ADMM algorithm. On the other hand, package loss during information exchange and errors in synchronization will cause the algorithm to converge at a non-optimal value. Hence, a trade-off between robustness, communication technology, computation resources and performance should be considered by grid operators before deciding which method to deploy.

V. CONCLUSIONS

Under the motivation of study different system partitioning for decentralized control, this paper provides a partitioning criteria and case study results for evaluating the influence of the cluster formation on the performance of the ADMM and KKT-based algorithms. The given methods are compared at the architecture, organizational and implementation level. The communication requirements for both methods are given with respect to the implementation for the decentralized solution to the DC-OPF problem. The classic ADMM aproach requires a larger number of iterations, the ADMM convergence is accelerated by the proposed adaptive penalty factor method. In the future, we intend to explore the dependence of the communication infrastructure on the decentralized scheme on different communication layers.

VI. ACKNOWLEDGMENTS

This work was financially supported by the Singapore National Research Foundation under its Campus for Research Excellence And Technological Enterprise (CREATE) program.

REFERENCES

[1] Y. Zhang and G. B. Giannakis, "Distributed Stochastic Market Clearing With High-Penetration Wind Power," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 895–906, mar 2016.

[2] Y. Zhang, M. Hong, E. Dall'Anese, S. Dhople, and Z. Xu, "Distributed Controllers Seeking AC Optimal Power Flow Solutions Using ADMM," *IEEE Transactions on Smart Grid*, vol. 3053, no. c, pp. 1–1, 2017.

[3] S. Hanif, H. B. Gooi, T. Massier, T. Hamacher, and T. Reindl, "Distributed congestion management of distribution grids under robust flexible buildings operations," *IEEE Transactions on Power Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[4] D. Kiran, A. R. Abhyankar, and B. K. Panigrahi, "Hierarchical clustering based zone formation in power networks," in *2016 National Power Systems Conference (NPSC)*, Dec 2016, pp. 1–6.

[5] M. Ma, L. Fan, and Z. Miao, "Consensus ADMM and Proximal ADMM for economic dispatch and AC OPF with SOCP relaxation," in *2016 North American Power Symposium (NAPS)*, no. 2. IEEE, sep 2016, pp. 1–6.

[6] P. N. Biskas and A. G. Bakirtzis, "A decentralized solution to the Security Constrained DC-OPF problem of multi-area power systems," in *2005 IEEE Russia Power Tech*. IEEE, jun 2005, pp. 1–7.

[7] T. Erseghe, "Distributed optimal power flow using ADMM," *IEEE Transactions on Power Systems*, vol. 29, no. 5, pp. 2370–2380, 2014.

[8] A. Kargarian, J. Mohammadi, J. Guo, S. Chakrabarti, M. Barati, G. Hug, S. Kar, and R. Baldick, "Toward distributed/decentralized DC optimal power flow implementation in future electric power systems," *IEEE Transactions on Smart Grid*, vol. 3053, no. c, 2016.

[9] J. Taylor, *Convex Optimization of Power Systems*, ser. Convex Optimization of Power Systems. Cambridge University Press, 2015.

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[11] S. Kar, G. Hug, J. Mohammadi, and J. M. F. Moura, "Distributed state estimation and energy management in smart grids: A consensus+ innovations approach," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 6, pp. 1022–1038, Dec 2014.

[12] C. Song, S. Yoon, and V. Pavlovic, "Fast admm algorithm for distributed optimization with adaptive penalty," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI'16. AAAI Press, 2016, pp. 753–759.

[13] B. S. He, H. Yang, and S. L. Wang, "Alternating Direction Method with Self-Adaptive Penalty Parameters for Monotone Variational Inequalities," *Journal of Optimization Theory and Applications*, vol. 106, no. 2, pp. 337–356, Aug 2000.

[14] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, vol. 5, Dec 2003, pp. 4997–5002 Vol.5.