Distributed Algorithm for SDP State Estimation

Yang Weng, *Student Member, IEEE*, Qiao Li, *Student Member, IEEE*, Rohit Negi, *Member, IEEE*, Marija Ilić, *Fellow, IEEE*

***Abstract*—Smart Grid State Estimation (SE) aims at providing robust and accurate system state estimate for subsequent control operations to accommodate the disturbance of highly intermittent components. Conventional SE for AC Power Grid formulates the estimation process as a non-convex optimization problem, which may reach a local optimal and stop. To compensate the drawback, Semidefinite Programming (SDP) was recently applied to convexify the non-convex problem with rank one relaxation technique, which shows perspectives with approximately globally optimal estimate. However, as the SDP approach is essentially a centralized algorithm, it is computationally expensive and non-robust to partial network failure, bad data. etc. To ameliorate the current SDP SE approach, this paper presents a distributed algorithm by employing Lagrangian method and graph theory results and particularly by dividing power networks in accordance with network 'cliques'. Correspondingly, significant improvements are illustrated in simulations on IEEE test beds.**

I. Introduction

With massive penetration of intermittent components such as green energy, traditional power grids nowadays are stressful in dealing with numerous uncertainties within the network. The modern smart grid construction and operation aims at transforming the traditional power network into a 'smart' one by using information and communication technology in an automatic, efficient way for power generation and distribution. One of the key intelligences of smart grid lies in its highly accurate state estimation (SE), which plays a core role in helping operators in a control center to monitor, analyze, and operate the grid. SE is also an important pre-requisite for many other crucial applications, such as contingency analysis [1], [2].

State estimation was first introduced by Schweppe [2]. For a nonlinear AC power system model, with or without bad data, SE is hard to solve due to the inherent non-convexity of the problem. The industry solution employs iterative optimization techniques, such as Iteratively Re-weighted Least Squares (IRLS) which is simply Newton's classic method, to compute a state estimate. However, because of the non-convex property, such method cannot guarantee the global optimal solution but only a locally optimal estimate.

Following recent advances in convexifying Optimal Power Flow (OPF) problems [3]–[5], where global optimal result over IEEE standard test beds and tree structure networks are found, lately, SE works try to reformulate the measurement model to convexify the SE problem with Semidefinite Programming (SDP) [6], [7]. [7] shows that SDP approach with rank one relaxation can significantly improve SE accuracy and provide an universal lower bound to characterize the duality gap.

While providing approximately globally optimal state estimate, SDP approach was observed to be computationally expensive [7], due to its optimization variable expansion from a vector variable to a more complex matrix variable. For example, the SDP method becomes increasingly time-consuming as the network size grows. Further, because of the centralized algorithm nature which requires full information gathering before computing, the SDP method is also non-robust to partial network failures. To remove such limits, this paper investigates and addresses a distributed algorithm to restructure the centralized SDP computation into local SDP problems without compromising the accuracy. To achieve restructuring, graph theory results on chordal graph are employed to separate the physical network, leading to a decomposition of the global constraint into local ones. Notice that [8] applies this technique for distributed OPF. Further, based on the cliques of the chordal graph, the SDP quadratic objective is further broken down by using the Lagrange dual method.

As a result, with message exchanges on coupling nodes between neighboring local networks, the original centralized SDP approach can be characterized in a distributed manner via local SDP SE computation. Obviously, a direct consequence is computational time reduction if parallel processors are used. Robustness is the second benefit. Via network decomposition, local SDP SE can be conducted along without message interchange with neighbors, which prevents local bad data or topology error propagation. Again, numerical results show significant reduction on computational time, fast rate of convergence, and robustness to local bad data, which is critical for power system analysis. Finally, to the authors' knowledge, this is the first paper proposing a distributed algorithm for SDP SE.

The rest of this paper is organized as the following: Section II-B introduces the system model and reviews the previous centralized SDP estimation approach; Section III explores the distributed algorithm; one toy example is illustrated in Section IV for the proposed algorithm; Section V demonstrates the simulation results and section VI concludes the paper; An Appendix on using power flow measurements in the distributed algorithm is also included.

Yang Weng and Qiao Li are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213 USA (e-mail: {yangweng}@andrew.cmu.edu).

Rohit Negi and Marija D. Ilić are professors at Carnegie Mellon University, Pittsburgh, PA, with the department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213 USA (e-mail: {negi, milic}@ece.cmu.edu).

II. Preliminaries

A. Model

The measurement model of the AC power system state estimation is expressed as follows [9]:

$$z_i = h_i(\boldsymbol{v}) + u_i \tag{1}$$

where the vector $\boldsymbol{v}$

$$\boldsymbol{v} = (|v_1|e^{j\delta_1}, |v_2|e^{j\delta_2}, \cdots, |v_n|e^{j\delta_n})^T$$

represents the power system states, u_i is the i^{th} additive measurement noise, which is assumed to be an independent Gaussian random variable with zero mean, i.e., $\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$, where Σ is a diagonal matrix, with the i^{th} diagonal element σ_i^2. z_i is the i^{th} telemetered measurement, such as power injection. Thus, $h_i(\cdot)$ is a nonlinear function for the i^{th} measurement. Note that in practice, the measurement set $\boldsymbol{z} = [z_1, z_2, \cdots, z_m]^T$ is usually made redundant.

The power system state estimation aims to find an estimate ($\hat{\boldsymbol{v}}$) of the true states ($\boldsymbol{v}$) that best fits the measurement set $\boldsymbol{z}$ according to the measurement model in (1), which is usually achieved by minimizing the following Weighted Least Square (WLS) criterion

$$\min_{\boldsymbol{v}} J_2(\boldsymbol{v}) = \left(\sum_{i=1}^{m} \left| \frac{z_i - h_i(\boldsymbol{v})}{\sigma_i} \right|^2 \right)^{\frac{1}{2}}. \tag{2}$$

B. Previous Work on SDP SE

The optimization problem in (2) is highly non-convex and difficult to solve optimally. In practice, the state estimation problem is conventionally solved by applying Newton's method, a local search algorithm which is extremely sensitive to the initial guess. To convexify the problem for global optimal SE, [6] and [7] transform the optimization variable from vector form into matrix form, followed by SDP with convex relaxation algorithms.

Here we review the SDP formulation in [7], but in a slightly different form, for the purpose of facilitating the distributed algorithm analysis in the next section. Specifically, if we define $W = vv^H$, then the following complex power injection model can be used in computing the state estimate.

$$\boldsymbol{s}_{\text{inj}} = \text{diag}\{\boldsymbol{v}\}\boldsymbol{i}^H = \text{diag}\{\boldsymbol{v}\boldsymbol{v}^H Y^H\} = \text{diag}\{WY^H\} \tag{3}$$

Further, define $M_{ik}^{\text{act}} = \frac{1}{2}(Y^H + Y)$ and $M_{ik}^{\text{rea}} = \frac{1}{2}(Y^H - Y)$, we have

$$\begin{aligned} \boldsymbol{p}_{\text{inj}} &= \text{diag}\{W(Y^H + Y)\} = \text{diag}\{M^{\text{act}}W\} \\ \boldsymbol{q}_{\text{inj}} &= \text{diag}\{W(Y^H - Y)\} = \text{diag}\{M^{\text{rea}}W\} \end{aligned}$$

As a result, the i^{th} real power injection $p_{\text{inj,i}} = \sum_{k=1}^{n} M_{ik}^{\text{act}} W_{ki}$, while the i^{th} reactive power injection $q_{\text{inj,i}} = \sum_{k=1}^{n} M_{ik}^{\text{rea}} W_{ki}$. As such, the corresponding SDP SE problem with power injection measurements is as follows:

$$\begin{aligned} \underset{W}{\text{minimize}} \quad & \sum_{i=1}^{n} \Big[(\sum_{k=1}^{n} M_{ik}^{\text{act}} W_{ki} - z_i^{\text{act}})^2 \\ & \qquad + (\sum_{k=1}^{n} M_{ik}^{\text{rea}} W_{ki} - z_i^{\text{rea}})^2 \Big] \\ \text{subject to} \quad & W \succeq 0. \end{aligned} \tag{4}$$

Notice that the power flow measurements will be discussed in the appendix A. For simplicity, in the next section, we focus on power injection measurements only.

III. Distributed Optimization

In this section, we will illustrate the decomposition with real and reactive power injection measurements.

A. Objective Decomposition

The SDP formulation (4) is equivalent to the following optimization, by adding equality constraints and auxiliary vector variables $\boldsymbol{y}^{\text{act}}$ and $\boldsymbol{y}^{\text{rea}}$.

$$\begin{aligned} \underset{W}{\text{minimize}} \quad & \sum_{i=1}^{n} (y_i^{\text{act}})^2 + \sum_{i=1}^{n} (y_i^{\text{rea}})^2 \\ \text{subject to} \quad & W \succeq 0, \\ & y_i^{\text{act}} = \sum_{k=1}^{n} M_{ik}^{\text{act}} W_{ki} - z_i^{\text{act}}, \\ & y_i^{\text{rea}} = \sum_{k=1}^{n} M_{ik}^{\text{rea}} W_{ki} - z_i^{\text{rea}}. \end{aligned}$$

Accordingly, we can write the lagrangian as below:

$$L(W, \boldsymbol{\lambda}^{\text{act}}, \boldsymbol{\lambda}^{\text{rea}}, \boldsymbol{y}^{\text{act}}, \boldsymbol{y}^{\text{rea}}) \tag{5}$$

$$= \sum_{i=1}^{n} \Big[(y_i^{\text{act}})^2 + \lambda_i^{\text{act}} (y_i^{\text{act}} - \sum_{k=1}^{n} M_{ik}^{\text{act}} W_{ki} + z_i^{\text{act}}) \Big] \tag{6}$$

$$+ \sum_{i=1}^{n} \Big[(y_i^{\text{rea}})^2 + \lambda_i^{\text{rea}} (y_i^{\text{rea}} - \sum_{k=1}^{n} M_{ik}^{\text{rea}} W_{ki} + z_i^{\text{rea}}) \Big] \tag{7}$$

And the dual problem is:

$$\begin{aligned} \max_{\boldsymbol{\lambda}} \min_{W,\ \boldsymbol{y}} \quad & \sum_{i=1}^{n} \Big[(y_i^{\text{act}})^2 + \lambda_i^{\text{act}} (y_i^{\text{act}} - \sum_{k=1}^{n} M_{ik}^{\text{act}} W_{ki} + z_i^{\text{act}}) \Big] \\ & + \sum_{i=1}^{n} \Big[(y_i^{\text{rea}})^2 + \lambda_i^{\text{rea}} (y_i^{\text{rea}} - \sum_{k=1}^{n} M_{ik}^{\text{rea}} W_{ki} + z_i^{\text{rea}}) \Big] \\ \text{subject to} \quad & W \succeq 0. \end{aligned}$$

The dual problem is equivalent to:

$$\begin{aligned} \max_{\boldsymbol{\lambda}} \min_{W,\ \boldsymbol{y}} \quad & \sum_{i=1}^{n} \Big[(y_i^{\text{act}})^2 + (y_i^{\text{rea}})^2 + \lambda_i^{\text{act}} (y_i^{\text{act}} + z_i^{\text{act}}) \\ & \qquad + \lambda_i^{\text{rea}} (y_i^{\text{rea}} + z_i^{\text{rea}}) \Big] - \sum_{i=1}^{n} \sum_{k=1}^{n} M_{ik} W_{ki} \\ \text{subject to} \quad & W \succeq 0. \end{aligned} \tag{8}$$

with $M_{ik} \triangleq \lambda_i^{\rm act} M_{ik}^{\rm act} + \lambda_i^{\rm rea} M_{ik}^{\rm rea}$.

As the objective has piecewise quadratic form over $y_i^{\rm act}$ and $y_i^{\rm rea}$ respectively, the optimal $y_i^{\rm act(rea)}$ can be obtained as presented below:

$$y_i^{\rm act(rea)} = -\frac{\lambda_i^{\rm act(rea)}}{2}$$

Thus, (8) turns to

$$\begin{aligned} \max_{\boldsymbol{\lambda}} \min_{W} \quad & \sum_{i=1}^{n} \Big[\lambda_i^{\rm act} (z_i^{\rm act} - \frac{\lambda_i^{\rm act}}{4}) \\ & + \lambda_i^{\rm rea} (z_i^{\rm rea} - \frac{\lambda_i^{\rm rea}}{4}) \Big] - \sum_{i=1}^{n} \sum_{k=1}^{n} M_{ik} W_{ki} \\ \text{subject to} \quad & W \succeq 0. \end{aligned} \tag{9}$$

Therefore, the objective is now linear with respect to the state matrix W. In the next section, we will introduce the positive semidefinite (PSD) constraint decomposition method for (4). Naturally, elements of W in the objective can be grouped in the same way for distributed computation.

Remark III.1. *The problem of an incomplete measurement set arises when measurements are not available on all buses, due to initial power grid planning, missing data, or communication errors, etc. To deal with such an incidence, with the measurement z_i unavailable, we can assume z_i to be the same as the estimated measurement in each iteration. One can also try to employ pseudo-measurements if they (measurements) are available.*

B. Constraint Decomposition

In this section, we will decompose the PSD constraint over W of (4) into equivalent submatrices PSD, according to [8], [10]–[12].

1) Graph Basics: Some graph-theoretic concepts necessary for subsequent discussion are introduced here. Particular emphasis is laid on chordal graphs. A typo in [8] was confirmed by one author and was also corrected in Theorem III.2.

An undirected graph is denoted by $G(V, E)$ with V to be the vertex set, and $E \in V \times V$ to be the set of edges. It is assumed, in this paper, that a graph has no self-loops, that is, $(v, v) \notin E$ for any $v \in V$. Two vertices $u, v \in V$ are defined to be adjacent if $(u, v) \in E$. A graph is considered as a complete graph if every pair of vertices is adjacent. For a subset $V' \subseteq V$, an induced subgraph $G(V', E')$ is with edge set $E' = E \cap (V' \times V')$. A clique of a graph is an induced subgraph which is complete. Formally, a clique is a set of pairwise adjacent vertices and a clique is maximal if its vertices do not constitute a proper subset of another clique. In our succeeding discussions, we call $C \subseteq V$ a clique of $G(V, E)$ whenever it induces a clique of $G(V, E)$. Further, a chord in a cycle is an edge connecting two non-consecutive vertices of the cycle. Therefore, a graph $G(V, E)$ is said to be chordal, or equivalently triangulated, if every cycle of length exceeding four has a chord.

2) PSD decomposition: Define a partial symmetric matrix W with element w_{ij} based on the graph $G(V, F)$ for some $F \in E$, where w_{ij} is defined if and only if edge $\{i, j\} \in F$. As a result, only parts of the matrix entries are specified. A completion of a partial symmetric matrix W is to find a PSD matrix which is a completion of a given partial symmetric matrix W. Denote by $\{C_r \subseteq V : r = 1, 2, \cdots, l\}$ the family of all maximal cliques of $G(V, F)$. An obvious necessary condition for W to have a PSD matrix completion is that each $W_{C_r C_r}$ is PSD ($r = 1, 2, \cdots, l$), where all the entries of the submatrix $W_{C_r C_r}$ are specified. We refer to the above results as the clique-PSD condition.

Theorem III.2. *(Grone et al. [10]) Any partial symmetric matrix W with specified entries $(i, j) \in F$ representing a chordal graph, has a positive semidefinite matrix completion, if and only if all submatrices $W_{C_r, C_r} (r = 1, 2, \cdots, l)$ are all PSD.*

Remark III.3. *Noted that [8] mistakenly states the above theorem by dropping the term of completion, which plays the core part of recovering W and subsequently the $\boldsymbol{v}$.*

3) Generate Chordal Graph: From the last subsection, we know that the PSD constraint decomposition can only work with chordal graph. Fortunately, some other works, such as [11], illustrate how to apply the chordal graph result for non-chordal graph. The key idea is about fill-in, or a triangulation of a graph G by adding virtual edges. In other word, one would like to fill-in the unknown matrix terms with respect to the graph, such that cliques are generated. Therefore, it is extremely important to determine a chordal extension which has the smallest number possible since this number directly affects the performance. Unfortunately, the problem of finding such an ordering that minimizes the fill-in is $\mathcal{NP}$ complete [11]. Hence, it seems reasonable at least in practice to employ some existing heuristic methods to obtain an ordering which possibly produces lesser fill-in [11].

On obtaining the chordal graph, one needs to search all maximal cliques. With the tool of perfect elimination ordering [13] one can efficiently locate all maximal cliques of a chordal graph, in linear time with respect to the vertex and edge number of the graph, where an elimination ordering over the graph is simply a numbering of the vertices with integers from 1 to n.

C. Algorithm summary

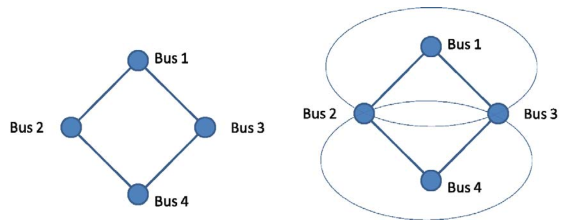
By using the PSD decomposition above to decompose the linear objective over W in (9), we can compute the SE in a distributed way, resulting in the following theorem.

Theorem III.4. *Optimization of (4) is the same as the following distributed algorithm:*

- *Step 1: Fix $\boldsymbol{\lambda}^{\rm act(rea)}$ and the coupling variables in W. For each clique, compute its local SDP SE for independent variables in W.*
- *Step 2: Fix $\boldsymbol{\lambda}^{\rm act(rea)}$ and the independent variables in W. Compute SDP SE for coupling variables in W.*
- *Step 3: Fix W. Update $\boldsymbol{\lambda}$ with standard algorithms.*
- *Step 4: If not converge, go to Step* 1.

IV. Illustrative Examples

We are using the IEEE 4 bus system (Fig.1) to illustrate the algorithm above. In this example, we assume that all four buses are equipped with active power injection measurements.



(a) Graph for IEEE 4bus System (b) Filled-in Chordal Graph

Fig. 1. Chordal Graph Generation

A. Preparation

Firstly, we generate the matrices $M^{\text{act}} = \frac{1}{2}(Y + Y^H)$.

$$M^{\text{act}} = \begin{bmatrix} 8.9852 & -3.8156 & -5.1696 & 0 \\ -3.8156 & 8.9852 & 0 & -5.1696 \\ -5.1696 & 0 & 8.1933 & -3.0237 \\ 0 & -5.1696 & -3.0237 & 8.1933 \end{bmatrix}$$

To make the graph chordal, we add a fill-in edge between buses 2 and 3, resulting in two cliques: $\mathcal{C}_1 = \{1, 2, 3\}$ (i.e., clique 1 includes bus $1, 2$ and 3.) and $\mathcal{C}_2 = \{2, 3, 4\}$. Based on M^{act}, we can define matrices W_1 and W_2 as follows:

$$W_1 = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & x_{11} & x_{12} \\ w_{31} & x_{21} & x_{22} \end{bmatrix}$$

$$W_2 = \begin{bmatrix} x_{11} & x_{12} & w_{24} \\ x_{21} & x_{22} & w_{34} \\ w_{42} & w_{43} & w_{44} \end{bmatrix}$$

where x_{ij} denotes a coupling variable, w_{ij} denotes an independent local variable. Further, we define submatrices M_1 and M_2 according to $\mathcal{C}_1$ and $\mathcal{C}_2$:

$$M_1 = \begin{bmatrix} 8.9852 & -3.8156 & -5.1696 \\ -3.8156 & 0 & 0 \\ -5.1696 & 0 & 0 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 0 & 0 & -5.1696 \\ 0 & 0 & -3.0237 \\ -5.1696 & -3.0237 & 8.1933 \end{bmatrix}$$

B. Computation

Step 1. Fix $\boldsymbol{\lambda}$ and coupling variables x_{ij}. Compute the local SDP over clique 1 and 2 separately with respect to (9) for independent variables.

$$\begin{aligned} \min_{W} \quad & -\sum_{i=1}^{3}\sum_{k=1}^{3} M_{1,ik} W_{1,ki} \\ \text{subject to} \quad & W_1 \succeq 0,\ W_1 \text{ is hermitian},\ \mathrm{x_{i,j}} \text{ is fixed.} \end{aligned} \tag{10}$$

$$\begin{aligned} \min_{W} \quad & -\sum_{i=1}^{3}\sum_{k=1}^{3} M_{2,ik} W_{2,ki} \\ \text{subject to} \quad & W_2 \succeq 0,\ W_2 \text{ is hermitian},\ \mathrm{x_{i,j}} \text{ is fixed.} \end{aligned} \tag{11}$$

Step 2. Fix $\boldsymbol{\lambda}$ and independent variables w_{ij}. Compute the optimization of (9) for coupling variables

$$\begin{aligned} \min_{W} \quad & -\sum_{i=1}^{4}\sum_{k=1}^{4} M_{ik} W_{ki} \\ \text{subject to} \quad & W \succeq 0,\ w_{i,j} \text{ is fixed.} \end{aligned} \tag{12}$$

Step 3. Fix W, solve (9) for $\boldsymbol{\lambda}$.
Step 4. If the optimization does not converge, go to step 1.

V. Simulation Results

In this work, simulations are implemented on the IEEE 4-Bus, 14-Bus, 30-Bus, 39-Bus 57-Bus and 118-Bus test systems.

A. Data generation

The data have been pre-processed by using the MATLAB Power System Simulation Package (MATPOWER) [14] [15]. Basically, we solve the power flow equations first in MATPOWER, which produces the true measurements without noise. Such measurements include the power injection on each bus and the transmission line power flows. Then we sample the measurements to create a measurement set large enough to ensure system observability.

B. Performance Comparison

To evaluate the performance, we compare the sum square errors among centralized SDP, decentralized SDP and traditional Newton's method. We conduct such a comparison with the IEEE 14 bus system for multiple times. Fig.2 illustrates 30 simulation results with the x coordinate as the test number, and the y coordinate as the metric of the sum square error. Evidently our proposed distributed algorithm essentially provides the same performance improvement as the centralized SDP method in [7]. The slightly bigger error for the distributed algorithm may result from convergence conditions.

C. Computational time speed up

The following shows the comparison of computational time in centralized SDP, parallel SDP, and traditional WLS method.

The simulation indicates that such a distributed algorithm for parallel computing can accelerate the SDP computation time for large system. Thanks to the fast rate of convergence, in the distributed computational part, much more time is saved than the time used to compute the dual variable $\boldsymbol{\lambda}$. From the figure, it is observed that although distributed SDP is consistently more time-consuming than the traditional WLS method, it dramatically reduces the time over the centralized SDP by flattening the curve. For small systems, it is also observed that the parallel computing method requires more

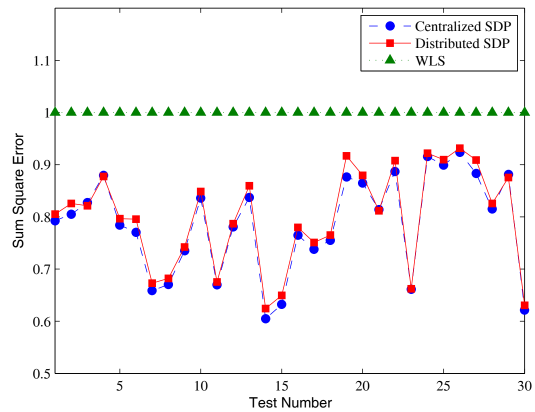


Fig. 2. Accuracy Comparison.

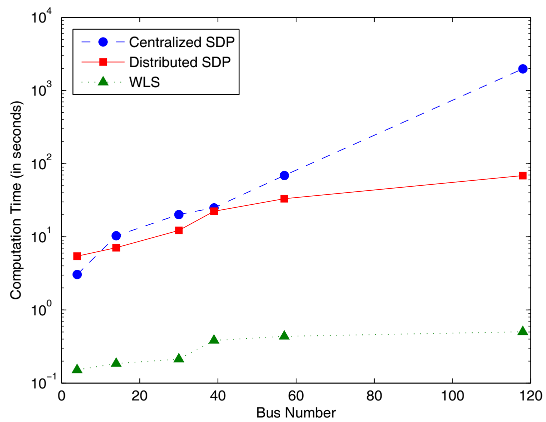


Fig. 3. Computational time comparison.

time than the distributed one. This is because, for a small system, dividing the network into multiple groups cannot substantially lower the optimization variable number. Instead, the computational time from the dual variable may produce a heavy burden to the algorithm as a whole. In summary, Fig.3 suggests distributed computation is better for large systems, while centralized computation is better for small systems.

D. Robustness to bad data

In this section, we consider the chordal graph division of the network as a tool, enabling local state estimation without information exchange on the bordering nodes. Therefore it is immune to bad measurements in other parts of the network.

For simulation, we add one bad data point into the active power injection on bus 4 of the IEEE 14 bus system as plotted in Fig.4. The goal is to test and compare centralized SDP SE and local SE without coordination. Table I shows the active power measurement residual (APMR) for the two methods. Apparently, if centralized SDP is adopted, when bad data appears in the network, error can propagate through neighboring local networks to deteriorate state estimates in the other areas. However, if one conducts local SDP, the estimate will be much more accurate than the centralized approach, leading to significant risk reduction when bad data or topology changes occur in parts of the grid.

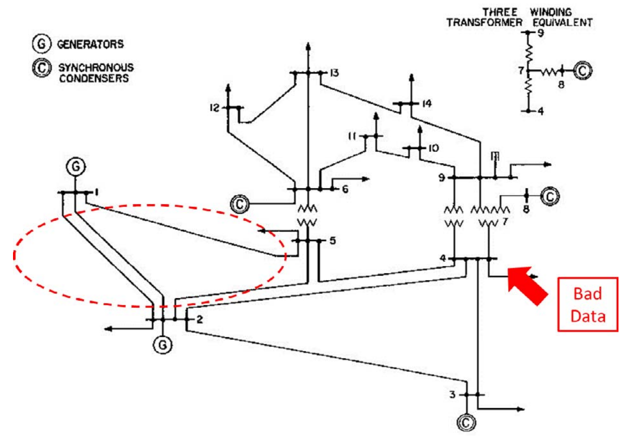


Fig. 4. Computational time comparison.

TABLE I
MEASUREMENT RESIDUALS

Method	Bus 1 APMR	Bus 2 APMR	Bus 5 APMR
Centralized SDP	0.024	0.017	0.064
Local SDP	0.006	0.007	0.007

VI. Conclusion

Recent advances in the Semidefinite Programming approach attempt to convexify the AC power system state estimation problem, which has been a major challenge for a long time due to its inherent non-convexity. While providing the approximately global optimum estimate, the SDP approach is non-robust to partial network failures and is computationally expensive, due to its centralized algorithm nature. To deal with such problems, this paper proposes a distributed SDP method based on graph theory, via dividing the large power grid into much smaller groups. Significant improvements in performance are obtained in terms of both time and robustness.

Acknowledgment

This work was supported in part by US NSF awards 0931978, 0831973 and 0347455.

References

[1] A. Wood and B. Wollenberg, *Power generation, operation, and control*, 2nd ed., 1996.

[2] F. C. Schweppe, J. Wildes, and D. B. Rom, "Power system static state estimation, parts 1, 2, 3." *IEEE Transactions on Power Apparatus and Systems*, p. 120, Jan. 1970.

[3] X. Bai, H. Wei, K. Fujisawa, and Y. Wang, "Semidefinite programming for optimal power flow problems," *International Journal of Electric Power & Energy Systems*, vol. 30, no. 6–7, pp. 383–392, Feb. 2008.

[4] J. Lavaei and S. H. Low, "Zero duality gap in optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 92–107, Feb. 2012.

[5] B. Zhang and D. Tse, "Geometry of feasible injection region of power networks," *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1508–1515, Sep. 2011.
[6] H. Zhu and G. B. Giannakis, "Estimating the state of ac power systems using semidefinite programming," *Proceedings 43rd North America Power Symposium (NAPS)*, pp. 1–7, Aug. 2011.
[7] Y. Weng, Q. Li, R. Negi, and M. D. Ilić, "Semidefinite programming for power system state estimation," *IEEE Power and Energy Society General Meeting*, Jul. 2012.
[8] A. Y. S. Lam, B. Zhang, and D. Tse, "Distributed algorithms for optimal power flow problem," Submitted for publication. [Online]. Available: http://arxiv.org/abs/1109.5229
[9] A. Abur and A. G. Exposito, "Power system state estimation: Theory and implementation," *CRC Press*, Mar. 2004.
[10] R. Grone, C. R. Johnson, E. M. Sa, and H. Wolkowicz, "Positive definite completions of partial hermitian matrices," *Linear Algebra and Applications*, vol. 58, pp. 109–124, 1984.
[11] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, "Exploiting sparsity in semidefinite programming via matrix completion i: General framework," *Journal of Mathematical Analysis and Applications*, vol. 11, no. 3, pp. 647–674, 2000.
[12] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima, and K. Murota, "Exploiting sparsity in semidefinite programming via matrix completion ii: Implementation and numerical results," *Mathematical Programming*, vol. 95, no. B, pp. 303–327, 2003.
[13] D. J. Rose, R. E. Tarjan, and G. S. Lueker, "Algorithmic aspects of vertex elmination on graphs," *SIAM Journal on Computing*, vol. 5, pp. 266–283, 1976.
[14] R. D. Zimmerman, C. E. Murillo-Sanchez, , and R. J. Thomas, "Matpower's extensible optimal power flow architecture," *IEEE Power and Energy Society General Meeting*, pp. 1–7, Jul. 2009.
[15] R. D. Zimmerman and C. E. Murillo-Sanchez, "Matpower, a matlab power system simulation package," *http://www.pserc.cornell.edu/ matpower/manual.pdf*, Jul. 2010.

APPENDIX

This section discusses about how to use power flow measurements in the distributed algorithm.

A. Power flow measurements

Similar to the power injection expression (3), the power flow representation can be formulated as:

$$s_{ij} = v_i i_{ij}^H = v_i (Y_{\text{bra}} \boldsymbol{v})_k \tag{13}$$

where k indicates that branch ij is the k^{th} branch in the case file. As a result, the power flow vectors are:

$$\boldsymbol{s}_{\text{flow}} = \text{diag}\{T\boldsymbol{v}\boldsymbol{v}^H Y_f^H\} = \text{diag}\{T\boldsymbol{v}\boldsymbol{v}^H Y^H T^H\} \tag{14}$$
$$\boldsymbol{p}_{\text{flow}} = \text{diag}\{T\boldsymbol{v}\boldsymbol{v}^H M^{\text{act}} T^H\} \tag{15}$$
$$\boldsymbol{q}_{\text{flow}} = \text{diag}\{T\boldsymbol{v}\boldsymbol{v}^H M^{\text{rea}} T^H\} \tag{16}$$

where T is a deterministic matrix mapping the bus voltages onto the branch voltages. As a result, our problem turns from (4) to:

$$\begin{aligned}
\underset{W}{\text{minimize}} \quad & \sum_{i=1}^{l} \Big[(T_{i,:} M^{\text{act}} W T_{i,:}^T - z_i^{\text{act.pf}})^2 \\
& + (T_{i,:} M^{\text{rea}} W T_{i,:}^T - z_i^{\text{rea.qf}})^2 \Big] \\
\text{subject to} \quad & W \succeq 0.
\end{aligned}$$

where l is the branch power flow measurement number. $T_{i,:}$ is the i^{th} row of matrix T.

The SDP SE above is equivalent to the following optimization, by adding equality constraints and auxiliary variables $\boldsymbol{y}^{\text{act.pf}}$ and $\boldsymbol{y}^{\text{rea.qf}}$.

$$\begin{aligned}
\min_{W} \quad & \sum_{i=1}^{l} \Big[(y_i^{\text{act.pf}})^2 + (y_i^{\text{rea.qf}})^2 \Big] \\
\text{subject to} \quad & W \succeq 0, \\
& y_i^{\text{act.pf}} = T_{i,:} W M^{\text{act}} T_{i,:}^T - z_i^{\text{act.pf}} \\
& y_i^{\text{rea.qf}} = T_{i,:} W M^{\text{rea}} T_{i,:}^T - z_i^{\text{rea.qf}}
\end{aligned}$$

Write the lagrangian:

$$L(W, \boldsymbol{\lambda}^{\text{act.pf}}, \boldsymbol{\lambda}^{\text{rea.qf}}, \boldsymbol{y}^{\text{act.pf}}, \boldsymbol{y}^{\text{rea.qf}}) \tag{17}$$
$$= \sum_{i=1}^{n} \Big[(y_i^{\text{act.pf}})^2 + \lambda_i^{\text{act.pf}} (y_i^{\text{act.pf}} - T_{i,:} W M^{\text{act}} T_{i,:}^T + z_i^{\text{act.pf}}) \Big] \tag{18}$$
$$+ \sum_{i=1}^{n} \Big[(y_i^{\text{rea.qf}})^2 + \lambda_i^{\text{rea.qf}} (y_i^{\text{rea.qf}} - T_{i,:} W W^{\text{rea}} T_{i,:}^T + z_i^{\text{rea.qf}}) \Big] \tag{19}$$

Therefore the dual problem is:

$$\begin{aligned}
\max_{\boldsymbol{\lambda}} \min_{W,\, \boldsymbol{y}} \quad & \sum_{i=1}^{l} \Big[(y_i^{\text{act.pf}})^2 + (y_i^{\text{rea.qf}})^2 + \lambda_i^{\text{act}} (y_i^{\text{act}} + z_i^{\text{act}}) \\
& + \lambda_i^{\text{rea}} (y_i^{\text{rea}} + z_i^{\text{rea}}) - T_{i,:} W M^{\text{act}} T_{i,:}^T \\
& - T_{i,:} W M^{\text{rea}} T_{i,:}^T \Big] \\
\text{subject to} \quad & W \succeq 0.
\end{aligned}$$

Notice that $T_{i,:} M^{\text{act}} W T_{i,:}^T$ can be converted into a simpler form as below with a new matrix M^{flow} such that

$$\sum_{i=1}^{l} T_{i,:} (M^{\text{act}} + M^{\text{rea}}) W T_{i,:}^T = \sum_{i=1}^{n} \sum_{k=1}^{n} M_{ik}^{\text{flow}} W_{ki}. \tag{20}$$

Consequently, the above optimization becomes equivalent to:

$$\begin{aligned}
\max_{\boldsymbol{\lambda}} \min_{W,\, \boldsymbol{y}} \quad & \sum_{i=1}^{l} \Big[(y_i^{\text{act.pf}})^2 + (y_i^{\text{rea.qf}})^2 + \lambda_i^{\text{act}} (y_i^{\text{act}} + z_i^{\text{act}}) \\
& + \lambda_i^{\text{rea}} (y_i^{\text{rea}} + z_i^{\text{rea}}) \Big] - \sum_{i=1}^{n} \sum_{k=1}^{n} M_{ik}^{\text{flow}} W_{ki} \\
\text{subject to} \quad & W \succeq 0.
\end{aligned}$$

Therefore, one can also conduct distributed SDP state estimation with a mere power flow measurement. Finally, it is also easy to combine the power flow measurements together with the power injection measurements in the main text of this paper.