

Graph Computing Based Distributed Parallel Power Flow for AC/DC Systems with Improved Initial Estimate

Wei Feng, Chen Yuan, Qingxin Shi, Renchang Dai, Guangyi Liu, Zhiwei Wang, and Fangxing Li

Abstract—The sequential method is easy to integrate with existing large-scale alternating current (AC) power flow solvers and is therefore a common approach for solving the power flow of AC/direct current (DC) hybrid systems. In this paper, a high-performance graph computing based distributed parallel implementation of the sequential method with an improved initial estimate approach for hybrid AC/DC systems is developed. The proposed approach is capable of speeding up the entire computation process without compromising the accuracy of result. First, the AC/DC network is intuitively represented by a graph and stored in a graph database (GDB) to expedite data processing. Considering the interconnection of AC grids via high-voltage direct current (HVDC) links, the network is subsequently partitioned into independent areas which are naturally fit for distributed power flow analysis. For each area, the fast-decoupled power flow (FDPF) is employed with node-based parallel computing (NPC) and hierarchical parallel computing (HPC) to quickly identify system states. Furthermore, to reduce the alternate iterations in the sequential method, a new decoupled approach is utilized to achieve a good initial estimate for the Newton-Raphson method. With the improved initial estimate, the sequential method can converge in fewer iterations. Consequently, the proposed approach allows for significant reduction in computing time and is able to meet the requirement of the real-time analysis platform for power system. The performance is verified on standard IEEE 300-bus system, extended large-scale systems, and a practical 11119-bus system in China.

Index Terms—AC/DC system, distributed parallel computing, graph computing, initial estimate, power flow analysis.

I. INTRODUCTION

THE rapidly-increasing capacity of modern power systems for power transmission has resulted in a pressing need to build large-scale alternating current (AC)/direct cur-

rent (DC) hybrid systems with high-voltage DC (HVDC) transmission lines. The needs for high voltage level (over ± 500 kV) and large DC power (over 3000 MW) in ultra-high voltage power systems [1] have led to the planning and implementation of more large-scale projects [2], [3]. The complexity and size of interconnected systems are also increasing. In addition to the physical challenges of such large systems, the DC links also complicate the power flow analysis, which is one of the most fundamental functions in an energy management system (EMS). The EMS must also be able to effectively deal with a large-scale hybrid system, providing accurate results faster than ever, to enhance the capability of system monitoring and the security of system operation [4], [5]. There are two widely-used approaches for AC/DC power flow analysis, namely, the unified method and the sequential method [6], [7].

1) In the unified method, the variables and equations of AC and DC grids are calculated simultaneously.

2) In the sequential method, the AC and DC equations are solved separately in a sequential process via inner and outer loops.

The most obvious advantage of the sequential method is the flexibility of implementing DC power flow analysis in the existing mature AC analysis, while, in the unified method, the solver must be re-programmed to accommodate the DC network. Thus, the sequential method is more flexible and suitable for solving large-scale hybrid systems. However, this method can be time-consuming because it calculates AC and DC networks repeatedly until achieving a global convergence, making it difficult to use in a real-time EMS, especially when the system is extremely large.

To reduce the computing time of the sequential method, there are two methods to be considered in the existing literature. The first method focuses on reducing the total amount of iterations. In [8], the reactive power injections are modeled in terms of the AC voltage and participate in the AC power flow analysis. Hence, the improved method converges to a solution in fewer iterations. In [9], power losses are modeled and updated along with the calculation of power exchange between AC and DC grids. Thus, the accuracy of each loop is improved, and fewer iterations are needed. In [10], some extreme high-load cases leverage virtual buses and optimal multipliers to guarantee the convergence in limited calculation times. Since the initial estimate of the New-

Manuscript received: July 7, 2018; accepted: May 9, 2019. Date of Cross-Check: May 9, 2019. Date of online publication: July 30, 2020.

This work was supported by the State Grid Corporation Technology Project (No. 5455HJ180022).

This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>).

W. Feng, Q. Shi, and F. Li (corresponding author) are with the University of Tennessee 1525 Coleman Road, Knoxville, Tennessee 37909, USA, and W. Feng is also with Global Energy Interconnection Research Institute North America, San Jose, California 95134, USA (e-mail: wfeng4@vols.utk.edu; qshi1@vols.utk.edu; fli6@utk.edu).

C. Yuan, R. Dai, G. Liu, and Z. Wang are with Global Energy Interconnection Research Institute North America, San Jose, California 95134, USA (e-mail: chen.yuan@geirina.net; renchang.dai@geirina.net; guangyi.liu@geirina.net; zhiwei.wang@geirina.net).

DOI: 10.35833/MPCE.2019.000047



ton-Raphson method is difficult to obtain due to the coupling of hybrid systems, the calculations have to be performed alternately in many times in the sequential method until global convergence.

The second method aims at improving the computing speed of power flow analysis. Some parallel approaches for solving a large sparse matrix, improving the Newton-Raphson method, and expediting data processing are proposed in [11]-[13]. Results indicate that parallel computing can help speed up the power flow analysis. Nevertheless, peripheral but time-consuming functions such as building matrices/vectors, checking convergence, and updating internal data are not addressed. Thus, there is a need to develop an efficient and holistic parallel or distributed method which can be used in all steps of AC/DC hybrid power flow analysis.

In the process of solving large-scale data-driven problems, graph computing has shown its parallel efficiency due to the protogenetic parallel database in modeling problems, and the underlying parallel algorithms based on vertex/edge [14]-[16]. Significant computing improvements with the application of graph-based methods have been demonstrated in the research fields of high-efficiency computing, social network analysis, security enhancement, etc. [17] - [19]. In recent years, graph computing has also been extended to solve power system problems such as distribution network reconfiguration, parallel power flow calculation, and real-time EMS development [20]-[22]. With the development of a graph database (GDB) and graph models for parallel computing, more complex problems can be entirely solved using graph-based methods and achieve better overall computing performance [23] - [25]. Therefore, graph computing can be utilized to form the foundation for high-efficiency computing.

A graph computing based distributed parallel implementation of the sequential method with an improved initial estimate approach to speed up the analysis of large-scale AC/DC systems without compromising accuracy is presented in this paper. First, the AC/DC system is modeled in a GDB. Then, it is divided into smaller AC grids by using graph partitions to disconnect DC links. Slack buses are provided for all power grids and used to synchronize the results. In addition, power flow via the corresponding DC lines are equivalently allocated as extra power injections. Hence, the AC power flow of the inner calculating loops in the sequential method can be performed parallelly. Meanwhile, by decoupling the AC/DC connections in an improved way, a good initial estimate can be achieved to reduce the total number of iterations in the sequential method. In addition, fast-decoupled power flow (FDPF) and data processing with node-based parallel computing (NPC) and hierarchical parallel computing (HPC) are employed to speed up both inner and outer calculating loops to speed up the entire calculating process.

The rest of this paper is organized as follows. Section II describes graph computing and its applications in solving power system problems. Section III presents the power flow models of hybrid AC/DC system and the general computing processes of the sequential method. Section IV proposes the improved initial estimate approach and explains how the proposed method is applied in the sequential method with im-

provements. Section V presents the test results and discusses the value of the proposed method, and Section VI concludes this paper.

II. GRAPH APPLICATIONS IN POWER SYSTEM ANALYSIS

A. Power System Modeling in a GDB

Graph theory is an advanced area of mathematical study which models complex networks and performs complicated calculations between the objects using “graphs”. Relevant graph-based techniques have been documented to significantly improve the general computing performance [26]. Based on graph theory, a system network can be modeled as a graph, $G(V, E)$. It consists of vertices representing all the objects in G collected in $V = \{V1, V2, \dots, Vn\}$, and edges denoting all the connections and relations between objects collected in $E = \{e_{i,j}\}$. Taking advantage of the pairwise relations of buses, branches and other attached devices, the graph model of a power system is intuitively built, and the graph model represents the system as shown in Fig. 1.

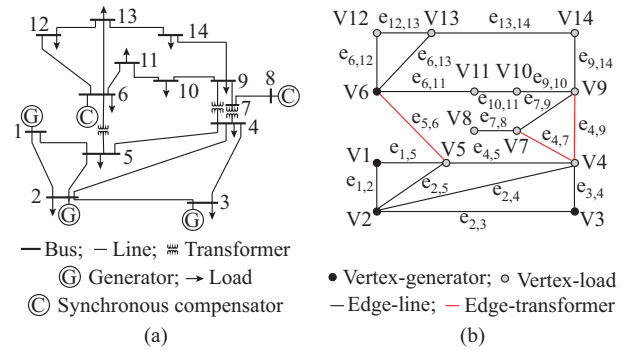


Fig. 1. Comparison of one-line and graph models for IEEE 14-bus system. (a) Single-line model. (b) Graph model.

The components constituting a conventional power grid can be classified as either vertex-attached or edge-attached. For example, generators, loads, static var compensators (SVCs) and converters together with their parameters are vertex-attached. The topology and attributes of transmission lines, transformers, filters and breakers are edge-attached. Thus, the graph model includes the information of both topology connections and related attributes. To illustrate the GDB in power system storage, the IEEE 5-bus system is used as an example.

As shown in Fig. 2, the attribute of connecting buses is used to establish the relationships between branches, generators, loads, and transformers in their corresponding tables. Meanwhile, intricate bridge tables are needed to provide data join functions with redundant storage elements in a relational database (RDB). Moreover, with the increase of data size and types, internal search gets complicated, significantly increasing the time consumption. Conversely, in the bottom plot of Fig. 2, the connectivity of different components is defined directly in an abstract graph. All of the unstructured attributes are classified and stored on vertices and edges. The bus-based parameters such as component type, generator capacity, reactive power limits, and load active/reactive power

can all be stored on the vertices. Similarly, the branch-based parameters such as line impedance, transformer ratio, and branch flow limits can all be stored on the edges. Since a GDB can reflect both connective relationships and attributes, operations on a GDB can be easier and faster. In [27], the query return tests on the system with over 10000 buses indicate that the GDB model can be 8 times faster than the conventional RDB model.

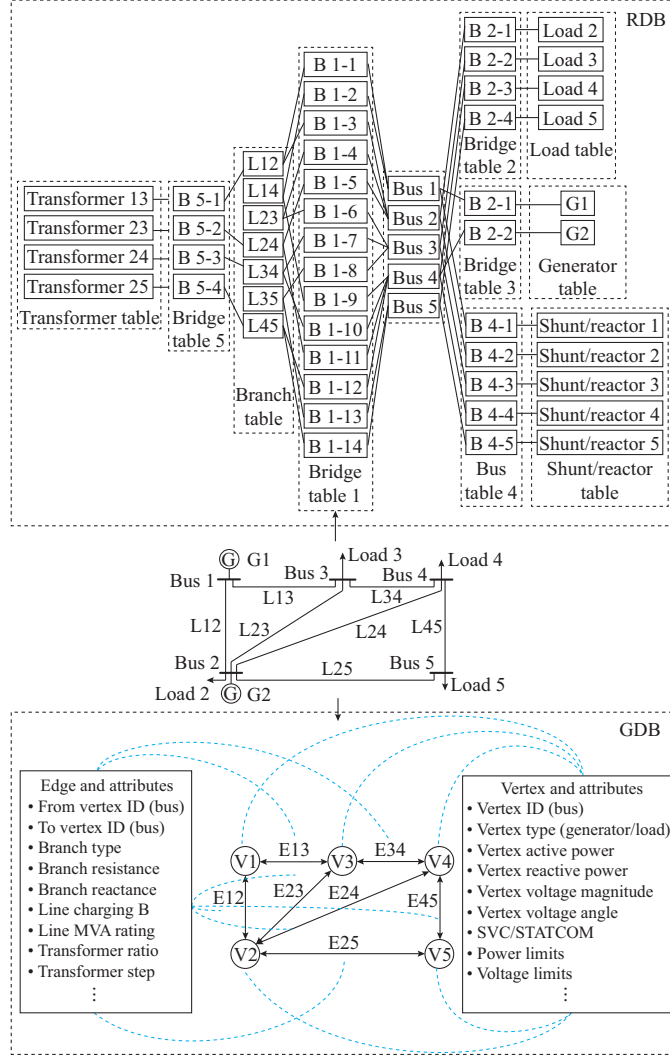


Fig. 2. Storage of power system modeling in two databases.

After modeling and storing a power system in a GDB, graph computing algorithms can be implemented and naturally executed on the underlying level with high parallel efficiency. The following subsections will introduce three graph-based algorithms and demonstrate their applications in solving power flow analysis.

B. Node-based Parallel Computing (NPC)

In graph theory, computing can be executed on vertices via graph traversal, which is a seamless integration of the GDB. Since each vertex is capable of conducting local computing independently with information from itself and its neighbors, it is feasible and efficient to divide one problem into smaller sub-problems, and then conduct parallel calculations on each vertex. One of the examples is matrix formulation with NPC.

As shown in Fig. 3 [21], the upper part illustrates the mapping relation of the expression of $Ax=b$ between graph computing and matrix formulation. “*” stands for the number of the specific but random bus in the system. In a graph model, the connections (edges) between vertices are symbolled as non-zero off-diagonal elements in matrix A . Similarly, the zero off-diagonal elements demonstrate that there exist no direct connections between vertices in the graph. Regarding the NPC method, giving an example of the admittance matrix, the off-diagonal elements are locally constructed according to the impedance of the corresponding edges. Meanwhile, each diagonal element is independently and locally calculated with the processing of impedance attributes at the corresponding vertex and its connected edges. Therefore, the whole admittance matrix is developed with only one-step graph computing and the values of all elements are calculated in parallel, which fits the GDB structure. The lower part gives the elaboration of the implementation of NPC. The workflow of the graph computing is constructed based on a bridging model of bulk synchronous parallel (BSP) [28]. The graph is initially stored in segments, each containing a set of vertices, together with the corresponding outgoing edges. The master thread subsequently assigns segments to the worker thread with respect to the CPU’s resources. For each worker, local computing is conducted in parallel. After performing the communication and synchronization, the results are transferred back to the master for final integration.

C. Hierarchical Parallel Computing (HPC)

Compared with NPC, which executes parallel calculations at each vertex, the HPC method assigns vertices to different levels based on the ordering of their connections. It performs parallel computing for vertices at the same level and solves the problem level by level in a hybrid sequential-parallel manner. One of the main applications is the LU factorization [25]. As shown in Fig. 4, three steps are involved to implement Cholesky factorization with HPC: ① determining fill-ins; ② forming an elimination tree; ③ partitioning the elimination tree via HPC. Different colors in Fig. 4 denote different levels of buses. In Fig. 4(a), $G(A_t)$ is the graph of matrix A_t , representing the matrix A at time t . Structural analysis in Fig. 4(b) exploits the vertex and edge distribution of $G(A_t)$, which is a non-zero pattern of A_t , to find a reordering of the vertices in the corresponding graph, such that the number of fill-ins can be reduced during matrix factorization. Meanwhile, in Fig. 4(c), a spanning tree $T(A_t)$, also known as an elimination tree, is created for LU factorization, and partitioned for parallelism. The vertices with the same color are grouped in the same level and can be computed simultaneously due to their independence to each other. In the graph model, the information of L_t and U_t is stored as an LU graph in $T(A_t)$ for problem solving.

Considering iterations in the sequential method for AC/DC hybrid systems, the LU factorization would be performed several times before the convergence. Thus, an efficient HPC algorithm is extremely helpful.

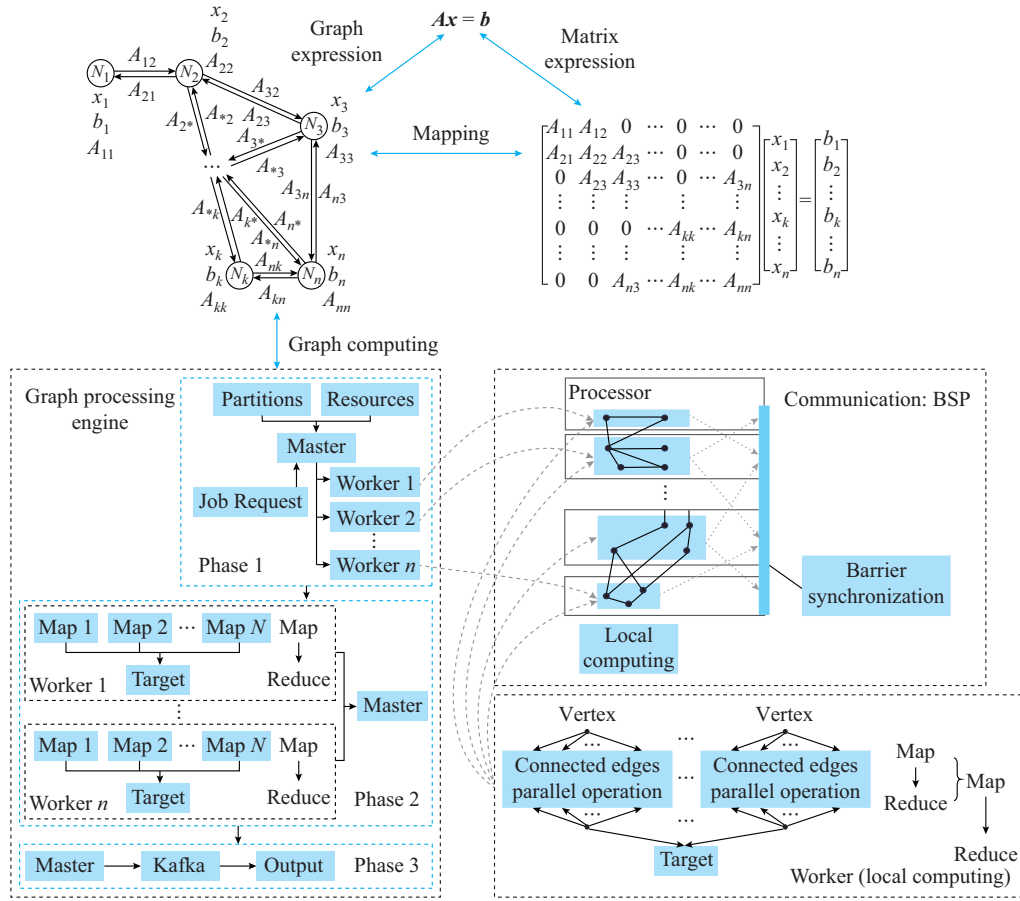


Fig. 3. Application of NPC in processing admittance matrix.

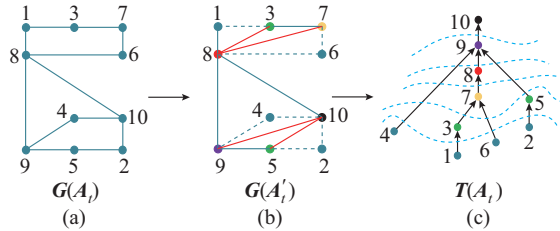


Fig. 4. Graph LU factorization. (a) Graph of matrix A_i . (b) Fill-ins determination. (c) Elimination tree and its partitions.

D. Distributed Computing with Graph Partition

The previous subsections have presented the concept and advantages of graph computing, but graph computing solves the system as an entity and performs the calculation and analysis together. Thus, all the vertices have to do the same computation locally in the beginning, then communicate and wait for synchronization in the end.

To expedite the calculation, the connecting topology of the hybrid system is taken into consideration to partition the original power network for distributed computing. Since most HVDC links are built to connect several disconnected AC grids crossing long distances, it is highly feasible to partition the network by disconnecting DC connections.

As shown in Fig. 5, there are two steps to perform the graph partition for the AC/DC system. To begin with, the DC line is disconnected from the original network, and the

coupled vertices $\{V4, V6\}$ are recorded as the initial searching points. Subsequently, simultaneous path searches start to check if the corresponding coupled points are still connected. For example, there are no paths between vertices $V4$ and $V6$ in Fig. 5(a), which means that the two AC grids are only connected via DC lines. In contrast, Fig. 5(b) denotes that the network cannot be split.

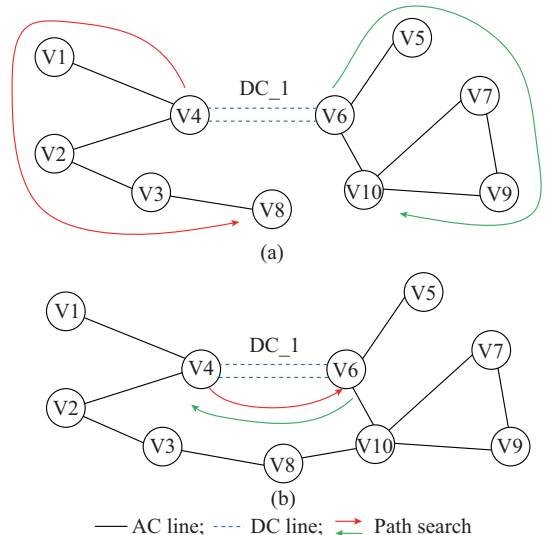


Fig. 5. Example of system partition by disconnecting DC lines. (a) Without AC connections between areas. (b) With AC connections.

E. Summary and Notes

Again, commercially-available graph computing packages built on top of GDBs have implemented parallel computing based on NPC and HPC to optimally utilize underlying multi-cores, which are commonly-available computing architecture today. Thus, end users can transparently apply these techniques in power flow analysis. By contrast, traditional parallel methods cannot automatically take advantage of multi-core computing architecture. Further, after system partitioning, the overall network is decomposed into several independent smaller AC grids, such that each grid can be solved in a multi-core architecture. This makes the sequential method for hybrid AC/DC power flow a natural fit for distributed computing.

These are the key reasons that the proposed approach will best utilize multi-core architecture and achieve significant performance acceleration with graph computing.

III. MODELING OF HYBRID AC/DC SYSTEMS FOR SEQUENTIAL METHOD

A. Modeling of Hybrid AC/DC Systems

The typical topology of DC links has been studied in detail. In general, the hybrid system can be modeled in three parts: AC grids, DC grids, and stations. AC grids are connected to stations via points of common coupling (PCC) and DC grids are connected via converters.

Figure 6 depicts a general equivalent power exchange model for an AC/DC system. In the simplified model, X_{si} and R_{si} are the equivalent reactance and resistance, respectively; U_{si} and θ_{si} are the voltage and angle at PCC, respectively; U_{ci} and θ_{ci} are the voltage and angle at the converter input point, respectively; I_{si} is the converter current; P_{si}/Q_{si} and P_{ci}/Q_{ci} are the power injections at PCC and the converter input point, respectively; and U_{dci} and I_{dci} are the voltage and current of the DC side, respectively.

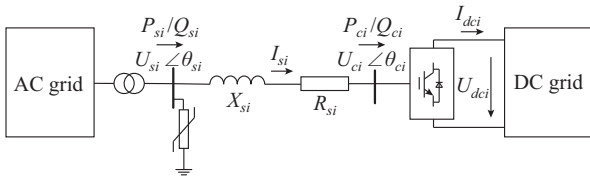


Fig. 6. Power exchange model of simplified AC/DC system.

In steady-state analysis, the equations of power balance can be utilized to describe the model [29]. First, in view of the AC side, power flow equations are given by:

$$\begin{cases} \Delta P_i = P_i^0 - U_i \sum_{j \in ii} U_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) - P_{si} \\ \Delta Q_i = Q_i^0 - U_i \sum_{j \in ii} U_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) - Q_{si} \end{cases} \quad (1)$$

where P_i^0 and Q_i^0 are the initialized or given active and reactive power on the buses, respectively; θ_{ij} is the angle difference between buses i and j ; G_{ij} and B_{ij} are the real and imaginary parts of the admittance matrix, respectively. In addition, the power injections at PCC can be calculated by:

$$\begin{cases} P_{si} = \frac{R_{si} U_{si} (U_{si} - U_{ci} \cos \delta_i) + X_{si} U_{si} U_{ci} \sin \delta_i}{R_{si}^2 + X_{si}^2} \\ Q_{si} = \frac{X_{si} U_{si} (U_{si} - U_{ci} \cos \delta_i) - R_{si} U_{si} U_{ci} \sin \delta_i}{R_{si}^2 + X_{si}^2} \end{cases} \quad (2)$$

where $\delta_i = \theta_{si} - \theta_{ci}$. Furthermore, the generalized power loss of the station can be modeled as:

$$P_{loss,i} = a_i + b_i I_{si} + c_i I_{si}^2 \quad (3)$$

where a_i , b_i , and c_i are the coefficients that can be achieved by field testing. For the DC side, since only active power can be transferred through DC lines, the power balance is given by:

$$P_{ci} = P_{dci} = U_{dci} I_{dci} \quad (4)$$

Beyond the equations of power balance, the relationship of converter voltage and DC voltage for the line commutated converter (LCC) can also be given by:

$$U_{dci} = N_r \left(\frac{3\sqrt{2}}{\pi} U_{ci} \cos \alpha_i - \frac{3}{\pi} X_{si} I_{dci} \right) \quad (5)$$

where N_r is the number of bridges; and α_i is the firing angle. Although the reactive power Q_{si} cannot be controlled directly by the LCC, the value can be calculated by:

$$\begin{cases} Q_{si} = P_{si} \tan \phi_i \\ \phi_i = \cos \alpha_i - \frac{X_{si} I_{dci}}{\sqrt{2} U_{si}} \end{cases} \quad (6)$$

where ϕ_i is the power factor. The stations can be equalized to PQ or PV buses depending on different control strategies.

B. Computing Processes of Sequential Method

As stated in the previous subsection, a typical hybrid system can be decoupled in three parts and then analyzed separately. Power flow analysis is performed in each part, and then the results at PCC and station side are compared and updated for the next calculation. By conducting this process in sequence, global convergence can be achieved when the results calculated from AC sides and DC sides equal out. The sequential method must complete power flow analysis in two loops as shown in Fig. 7.

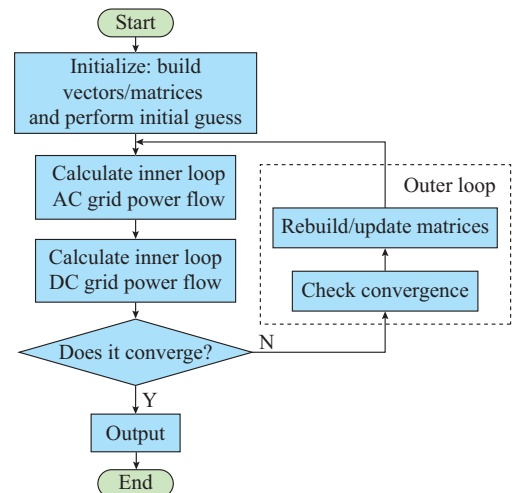


Fig. 7. Inner and outer loops of sequential method.

Since it is difficult to measure accurate injection power at PCC as AC and DC grid input, the inner and outer loops are performed repeatedly to upgrade the values until global convergence is achieved. The power flows for the AC and DC grids are performed in the inner loops. Subsequently, the global convergence and value update are executed in the outer loops. An integral computing cycle includes an inner loop and an outer loop. In general, the sequential method is flexible and effective for use with large-scale hybrid systems with complex DC connections. By decoupling the system, modifications, and changes on the DC side such as converter blocks, transformer adjustments, and the switch of operating strategies, will not affect AC power flow, which is the main part of the calculation. Due to the multiple iterations in the two calculating loops, however, this does increase computing time. Therefore, new methods and tools to reduce the total number of iterations and expedite the overall calculation are necessary.

IV. IMPROVEMENT AND IMPLEMENTATION OF PROPOSED COMPUTING APPROACHES

A. Improved Initial Estimate Approach

In the conventional sequential method, the uncertainties of power injections and power losses can greatly increase the total number of iterations before achieving global convergence. As discussed in Section III, power injections and losses at stations cannot be determined ahead of the calculation and should be calculated twice from both the AC and DC sides. After that, the results are compared and updated in the outer loops. Therefore, the calculation of the station values is independent of the inner loops of power flow analysis. Furthermore, since the convergence performance of the Newton-Raphson method highly depends on the initial estimate, if an initial estimate that is sufficiently close to the roots is given, the number of iterations can be greatly reduced. Therefore, the computing time can be significantly reduced if a good initial estimate can be provided instead of using a flat start.

Note that in Fig. 6, the differences of power injections at PCC and station input are primarily caused by the equivalent R_{si} and X_{si} . Since their values are very small compared with normal line resistance and reactance [30], the differences between P_{si} and P_{ci} can be initially ignored. Thus, the station can be modeled as an extra equalized AC bus connected to the AC grid as shown in Fig. 8.

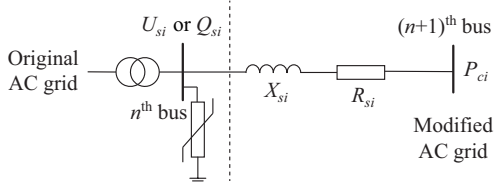


Fig. 8. Modified AC grid with new decoupled method.

Assuming that the original AC grid consists of n buses including PCC, by putting the station into the AC grid, the new modified AC grid is added with one branch and one bus afterwards. Thus, the new grid consists of $(n + 1)$ buses.

Furthermore, since P_{si} is approximately equal to P_{ci} , the new $(n + 1)^{\text{th}}$ bus can be set as a load which only absorbs active power. Consequently, the active power and power losses become parts of the AC grid and can be calculated together with AC power flow. In addition, since the topology of AC grids does not change, the results can be solved quickly via the FDPF method.

After the initial estimate is achieved, the main sequential method can be further expedited by implementing graph computing based techniques.

B. Implementation of Graph Computing Based Techniques in Sequential Method

The conventional sequential method can be time-consuming due to multiple cycles of power flow calculation, equation formulation, PCC values updates, etc. To speed up the overall calculation without altering the main algorithms or compromising accuracy, graph computing based methods including graph partitioning, NPC, and HPC are employed, as shown in the flowchart in Fig. 9, where A_n is the n^{th} partitioned area; α and γ are the firing and extinction angles, respectively; and ϕ_R and ϕ_I are the power factors in rectifier and inverter stations, respectively.

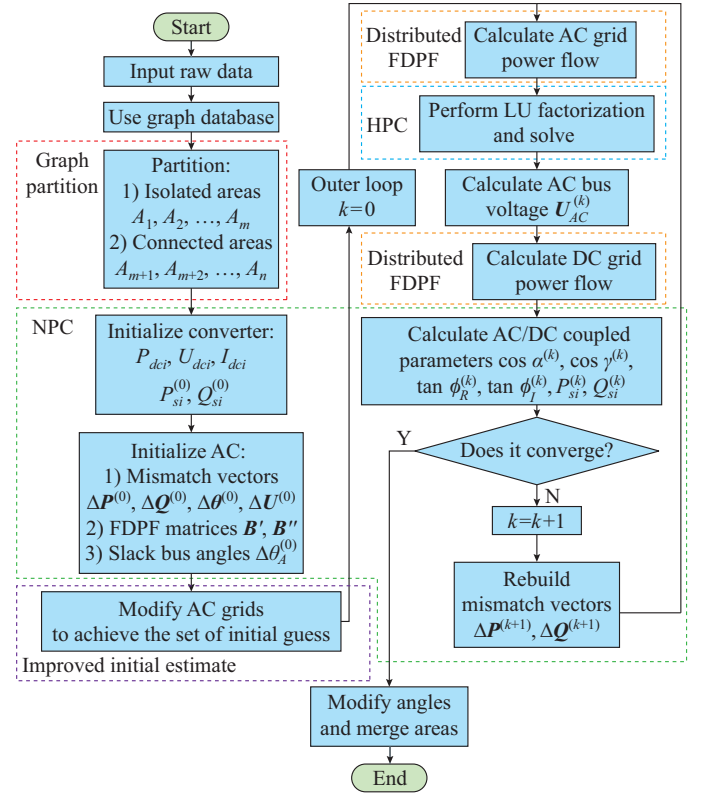


Fig. 9. Flow chart of graph computing based distributed parallel AC/DC power flow with improved initial estimate approach.

To further illustrate how the graph partition is employed, a general model is given in Fig. 10. First, graph partition is performed after the power system network is modeled and stored in the GDB. If we consider the topologies of the existing projects, a large-scale hybrid system can be partitioned into two types of sub-areas depending on the connection pattern, as described below.

1) Type 1: isolated areas with pure AC components. After disconnecting DC lines, the sub-area is only composed of AC components, and isolated from other areas such as Area 2 and Area 3.

2) Type 2: connected AC areas with tie-line. The disconnection of DC lines does not divide the connected areas. In practical projects, long-distance high-voltage alternating current (HVAC) and HVDC lines may coexist to enhance power transmission capacity. Thus, Type 2 can be joined, generating a new larger area such as in Area 1 in Fig. 10.

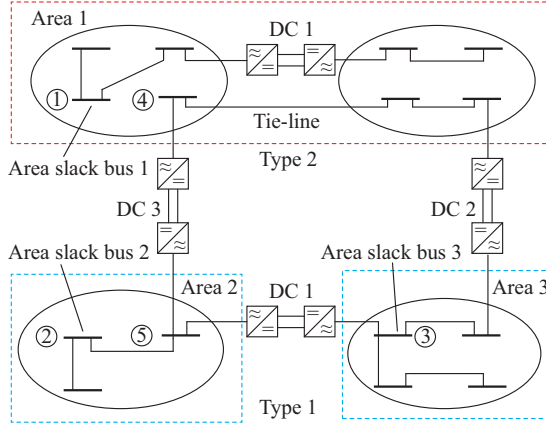


Fig. 10. Partition for generating distributed areas.

In the example of DC 3 branch, A_1 and A_2 are interconnecting and also connecting buses 4 and 5. During the process of partitioning, DC 3 branch is disconnected, and the power on DC converters are replaced by extra power injections at buses 4 and 5, i.e., $[P_{s4}, Q_{s4}, P_{s5}, Q_{s5}]$. The rest of the DC links are equivalently replaced as extra load injections. After the system is partitioned and equivalent power is injected, new area slack buses are selected for the isolated areas (buses 1, 2 and 3 in Fig. 10). In this way, all of the areas can be calculated simultaneously with a much lower computing burden compared with the size of the original problem. Then, FDPF can be employed as:

$$\begin{cases} \mathbf{B}'\Delta\theta = \Delta\mathbf{P}/|\mathbf{U}| \\ \mathbf{B}''\Delta\mathbf{U} = \Delta\mathbf{Q}/|\mathbf{U}| \end{cases} \quad (7)$$

where $\mathbf{B}'$ and $\mathbf{B}''$ are the sub-matrices of the approximated Jacobian matrix; $\Delta\mathbf{U}$ and $\Delta\theta$ are the incremental vectors of bus voltage magnitude and phase angle, respectively; and $\Delta\mathbf{P}/|\mathbf{U}|$ and $\Delta\mathbf{Q}/|\mathbf{U}|$ are the mismatching vectors of active and reactive power divided by corresponding bus voltage magnitude, respectively. The following text will further explain how NPC and HPC are implemented to improve the calculation.

1) NPC: formulating power flow equation (matrices and vectors), checking convergence and updating values. Considering that $\mathbf{B}'$ and $\mathbf{B}''$ are approximated Jacobian matrices, the diagonal elements represent the buses in the network and non-zero off-diagonal elements depict the existing connections between buses. In addition, each row vector is related to a corresponding bus in the system. In each row vector, non-zero off-diagonal elements are linked edges from the corresponding node and the diagonal element represents the node itself. Consequently, all the elements of the matrices

can be generated locally and independently, indicating the feasible application of NPC. The right-hand side vectors in (1) are updated similarly. Thus, all power flow equations can be formulated using NPC. With the exception of equation formulation, convergence checks and status updates can also be conducted locally.

2) HPC: solving power flow. The sequential method requires multiple processes of FDPF; thus, an efficient solver is able to quickly find the solution. As shown in Fig. 9, in each cycle, HPC-based FDPF is performed on distributed areas at the same time. Vertices at the same level are analyzed and calculated in parallel. Details have been discussed in Section II-C.

V. PERFORMANCE TESTING

A. Computing Environment and Main Test Cases

The experiments are performed on a Linux server, with installation of a programmable GDB platform, TigerGraph. The server has a Xeon E7-8867 (V3 2.50 GHz) CPU and 64 GB memory with 16 cores which will be the platform for implementing the distributed AC/DC power flow algorithm. The operation system is CentOS 6.8 and the GDB version is 2.2. To guarantee precision, the iterative tolerance of both inner and outer loops is set as 10^{-5} .

To verify the accuracy and demonstrate the computing performance of the proposed method, the standard IEEE 300-bus system, two extended South Carolina 500-bus systems [31], and a practical AC/DC system in China are tested. The two extended South Carolina systems have 6000 and 12000 buses, respectively, and the China system has 11119 buses. To illustrate the structure of the extended South Carolina systems, we exemplify the 6000-bus case in detail. The 6000-bus case is composed of 12 of the same South Carolina systems, formatting a loop network with 12 identical DC line connections with each area only connected to two neighboring grids. The structures of these three large-scale testing systems are listed in Table I. The DC parameters are given in Table II.

TABLE I
CONSTRUCTIONS OF LARGE-SCALE TEST SYSTEMS

Case	Topology
6000-bus	12 South Carolina grids/12 DC lines
12000-bus	24 South Carolina grids/24 DC lines
11119-bus	Hybrid grid in China/9 DC lines

TABLE II
PARAMETERS OF DC LINE IN IEEE 300-BUS SYSTEM

Type	Value
No. of bridges	4
DC power (MW)	100
DC voltage (kV)	460
Resistance (Ω)	6.8
Reactance (Ω)	6.2
T-ratio	0.7478
Firing angles ($^\circ$)	[15, 20]/[18, 20]

Compared with the extended ideal systems, which can be fully partitioned, the 11119-bus case is complicated by two factors. First, due to the coexistence of HVAC and HVDC in long-distance transmission, some areas are still connected

even after disconnecting DC lines. Second, the operation parameters of the DC lines are diverse from each other, as shown in Table III.

TABLE III
PARAMETERS OF DC LINES IN PRACTICAL 11119-BUS SYSTEM

No. of DC line	Rectifier bus	Inverter bus	Bridge	DC power (MW)	DC voltage (kV)	X/R (Ω)	Transformer ratio	α ($^\circ$)	γ ($^\circ$)
1	88	1003	2	750	250	3.995/2.1	0.9375	[16, 20]	[18, 20]
2	2957	4901	2	1500	500	7.936/4.0	0.9750	[15, 20]	[18, 20]
3	3003	3717	2	1500	500	7.936/4.0	0.9400	[15, 20]	[18, 20]
4	3520	4903	2	600	500	17.902/9.4	0.9500	[15, 20]	[16, 20]
5	3528	4900	2	1500	500	7.936/4.0	0.9750	[16, 20]	[16, 20]
6	7453	3746	2	360	225	9.261/3.6	0.8450	[18, 20]	[18, 20]
7	7455	9491	2	750	250	3.995/2.1	0.9750	[12, 20]	[12, 20]
8	7456	7772	2	1500	500	7.575/3.5	0.9375	[12, 20]	[15, 20]
9	8417	4902	4	2000	800	5.419/2.3	0.9500	[16, 20]	[18, 20]

To further illuminate the difference of structures, the admittance matrices of the 6000-bus case and the 11119-bus case are shown in Fig. 11. It should be noted that the 6000-bus case can be fully partitioned by disconnecting the DC links between adjacent AC areas as shown in Fig. 11(a). Additionally, the sizes of each partitioned AC area are equal and the parallel computing time for each area would be very close. However, the 11119-bus case has only five areas that can be fully partitioned as shown in Fig. 11(b), despite the existence of 9 DC lines. In addition, the largest partitioned area contains 3800 buses while the smallest one has only 480 buses. The influence of these differences on power flow analysis will be further discussed in the following subsections.

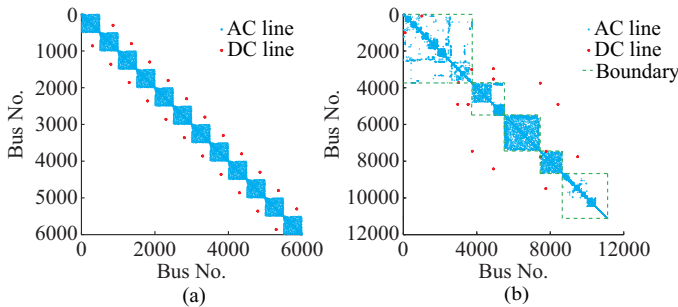


Fig. 11. Admittance matrices of different cases. (a) 6000-bus case. (b) 11119-bus case.

B. Accuracy Verification

The commercial software PSS/E (v 33.0) is used for accuracy comparison. Since the equivalent power injections from DC lines to AC grids are much larger than other normal loads, the results at the coupling AC/DC buses are most likely to have the biggest differences. The test results also indicate that the biggest differences between PSS/E and the proposed method occurs at the DC linked buses. Thus, we provide the power flow results of these buses, together with DC operation values, as shown in Tables IV and V.

TABLE IV
ACCURACY OF CALCULATED DC VALUES FOR IEEE 300-BUS SYSTEM

Platform	Bus	$ U $ (p.u.)	θ ($^\circ$)	α/γ ($^\circ$)
TigerGraph	119	1.04350	40.98768	16.240
	120	0.99818	37.72667	18.375
PSS/E	119	1.04350	40.98740	16.240
	120	0.99819	37.72660	18.379

TABLE V
OVERALL ACCURACY COMPARISON FOR LARGE-SCALE SYSTEMS

Case	Biggest differences compared with PSS/E		
	$ U $ (p.u.)	θ ($^\circ$)	α/γ ($^\circ$)
6000-bus	0.00001	0.0008	0.003
12000-bus	0.00001	0.0008	0.003
11119-bus	0.00003	0.0027	0.008

The biggest differences of voltage magnitude, phase angle, and triggering angle are 0.00001 p.u., 0.0006 $^\circ$, and 0.004 $^\circ$, respectively, which are negligible in practical applications. The comparisons for the large-scale systems with multiple DC lines are listed in Table V. It can be observed that for the systems with multiple DC lines and complex topologies, the proposed method still has satisfactory accuracy.

C. Computing Performance of Main Test Cases

To demonstrate the speedup in computing performance, four algorithms are compared, including the graph computing-based method (GC), the conventional sequential method (CONV), the default unified method and the modified sequential method in PSS/E (UPSS and SPSS). All the tests are performed on the same environment.

Table VI compares the computing speed of the four methods for large-scale systems. It should be declared that both the 6000-bus case and the 12000-bus case require two outer calculation loops for the sequential method, and that the 11119-bus case needs three outer loops. It can be seen that

for large-scale systems, the graph-based method can provide better performance compared with the conventional method. For the extended ideal systems, since the size of each partitioned area is the same, the speedup can be around 4.5. For the practical system in China, as its connections are more complex, the speedup is still over 3. Even compared with the UPSS, the improvement is still significant. Besides the overall speedup, the computing time of performing graph partitions, formulating power flow equations (including building matrices, vectors and updating interface data), and performing inner FDPF is also compared to indicate the detailed improvements in Fig. 12.

TABLE VI
COMPARISONS OF COMPUTING TIME WITH DIFFERENT METHODS

Case	Computing time (ms)			
	GC	CONV	UPSS	SPSS
6000-bus	76.135	308.2	247.6	295.2
12000-bus	126.180	563.4	453.2	587.3
11119-bus	239.100	743.1	621.5	765.8

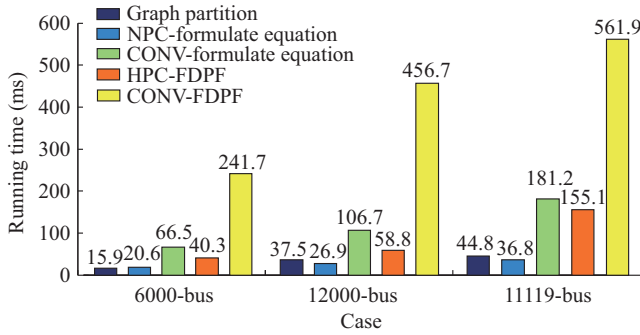


Fig. 12. Comparison of detailed running time of main calculating procedures.

As discussed previously, the major improvements are in the areas of graph partition, NPC, and HPC. Specifically, throughout the entire calculation, graph partitioning is only performed once at the beginning. A 10000-bus system can be completely calculated within 50 ms. Thus, the level of improvement justifies the extra cost.

Further, the computing time for formulating equations and performing FDPF is compared in Table VII.

TABLE VII
COMPARISON OF TIME SPENT ON TWO MAJOR PROCEDURES

Case	Computing time for formulating equations (ms)		Computing time for performing FDPF (ms)	
	GC	CONV	GC	CONV
6000-bus	10.3×2	33.2×2	20.1×2	120.9×2
12000-bus	13.8×2	53.4×2	29.4×2	228.4×2
11119-bus	12.3×3	60.4×3	51.7×3	187.3×3

Based on Fig. 12 and Table VII, the main features of graph computing can be analyzed. Firstly, the time spent on graph partitioning highly depends on the topological complexity of the hybrid systems. The parallel partition searches

only 500 buses in each area for both the 6000-bus case and the 12000-bus case. However, despite fewer DC lines, the longest search path in the 11119-bus case has to cover 3800 buses, which increases the computing time. Secondly, the speedup of NPC is affected by system scale. Since each vertex in the graph model can be seen as an independent logic and computing unit, the time saved is more significant on a larger system. Consequently, the speedup for grids with over 10000 buses can be as much as 4.9 compared with a speedup of 3.23 for a 6000-bus system. Thirdly, the performance of HPC is mainly affected by the number and size of partitioned areas. Since the two extended systems are ideally made up of the same-size areas, the computing time for each area is close. However, since the sizes of partitioned areas in the 11119-bus case vary widely, the overall computing time is determined by the largest area with 3800 buses. Therefore, the speedup is a little smaller compared with the 6000-bus and 12000-bus cases.

Another advantage of graph computing is the efficient utilization of multiple cores. The parallel algorithms based on vertices and edges can be programmed in the underlying multi-core architecture with a graph model. Thus, parallel cores can perform local parallel calculations directly. Figure 13 depicts the main computing time in one sequential calculating cycle with different numbers of cores in a single CPU. As the partition comparison shows, since the search range of partitioning has been narrowed to DC connections, even a single core is able to gain an ideal speedup. Conversely, as previously discussed, NPC can utilize more computing units to perform local calculation due to the independent calculating ability of each vertex. Hence, better performance can be achieved when more cores are in use, and the parallel efficiency of NPC is also demonstrated. With the use of eight cores, the speedup is satisfactory.

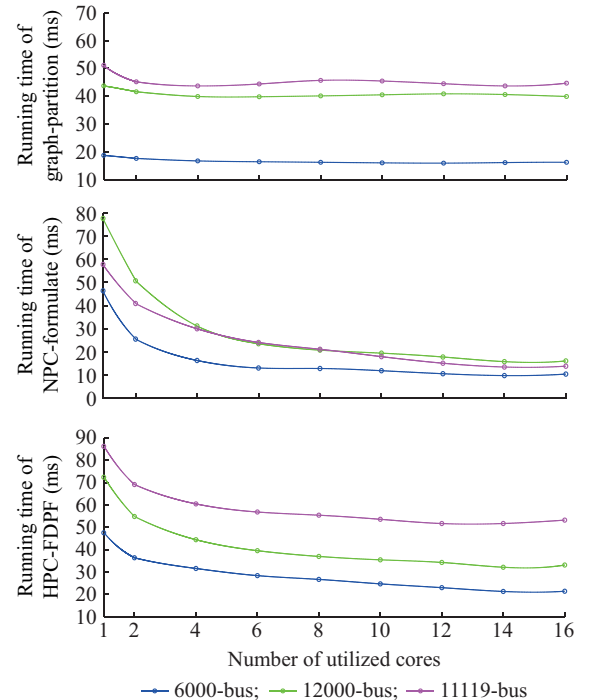


Fig. 13. Comparison of detailed computing time with multiple cores.

While the parallel performance of HPC can benefit from the use of a greater number of cores, the number and size of partitioned areas also affect the real computing speed. When the same partition areas have fewer buses, as in the 6000-bus and 12000-bus cases, more cores (~16) can be utilized to achieve better speedup. In the 11119-bus case, since the partitioned areas are fewer and the largest area has a much higher number of buses, the best performance is achieved by using 12 cores. In all cases, when a greater number of cores are put into use, the overhead time spent on allocating cores, reading memory, and transmitting data will also increase.

D. Relationship Between Speedup Ratio and System Size

As discussed in the previous sections, one of the advantages of a graph computing-based method is that each vertex in the graph model can be used as an independent logic and computing unit in the process of parallel computing. The proposed method performs better when used for larger systems, and it is therefore necessary to test the speedup ratios on larger systems. In this subsection, several extended large-scale cases based on South Carolina 500-bus systems are tested, and the computing time is compared using the CONV method. Note that these extended cases have similar connecting topologies as the 6000-bus and 12000-bus cases. The results are listed in Table VIII.

TABLE VIII
SPEEDUP RATIOS ON LARGER SYSTEMS

Case	Computing time (ms)		Speedup ratio
	GC	CONV	
6000-bus	76.135	308.2	4.048
12000-bus	126.200	563.4	4.464
24000-bus	263.430	1404.7	5.332
36000-bus	400.230	2629.9	6.571
48000-bus	556.190	4412.3	7.933

The comparison results in Table VIII indicate that the graph computing-based method has higher speedup ratios when the target hybrid systems are of larger sizes. The speedup ratio increases from 4.048 in the 6000-bus case to 7.933 in the 48000-bus case, which results in better computing performance. Therefore, the proposed method has great potential for the application in solving the power flow of large-scale systems.

VI. CONCLUSION

In this paper, a parallel power flow method using a GDB and graph computing methods are developed for hybrid AC/DC systems to improve computing performance. After the network is modeled and stored in a GDB, the vertex and edge based parallel computing can be performed naturally and directly. With graph partitioning, the original large-scale system can be split into multiple independent AC grids, which are fit for distributed processing. Further, the graph computing techniques based on NPC and HPC, which efficiently utilize the underlying multi-core computing architecture, are implemented in the sequential method for AC/DC

hybrid power flow to speed up the main time-consuming processes of formulating equations and solving FDPF. Simulation results on large-scale systems with over 10000 buses show significant speedup of the proposed method without compromising accuracy. Furthermore, the performances of multi-cores on graph partitioning, NPC and HPC also demonstrate high parallel efficiency.

REFERENCES

- [1] J. M. Maza-Ortega, E. Acha, S. García *et al.*, "Overview of power electronics technology and applications in power generation transmission and distribution," *Journal of Modern Power Systems and Clean Energy*, vol. 5, no. 4, pp. 499-514, Jul. 2017.
- [2] W. Long and S. Nilsson, "HVDC transmission: yesterday and today," *IEEE Power and Energy Magazine*, vol. 5, no. 2, pp. 22-31, Mar.-Apr. 2007.
- [3] T. An, C. Han, Y. Wu *et al.*, "HVDC grid test models for different application scenarios and load flow studies," *Journal of Modern Power Systems and Clean Energy*, vol. 5, no. 2, pp. 262-274, Mar. 2017.
- [4] N. M. Haleem and A. D. Rajapakse, "Application of new directional logic to improve DC side fault discrimination for high resistance faults in HVDC grids," *Journal of Modern Power Systems and Clean Energy*, vol. 5, no. 4, pp. 560-573, Jul. 2017.
- [5] P. Palensky and D. Dietrich, "Demand side management: demand response, intelligent energy systems, and smart loads," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 3, pp. 381-388, Aug. 2011.
- [6] J. Lei, T. An, Z. Du *et al.*, "A general unified AC/DC power flow algorithm with MTDC," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 2837-2846, Jul. 2017.
- [7] J. Beerten, S. Cole, and R. Belmans, "Generalized steady-state VSC MTDC model for sequential AC/DC power flow algorithms," *IEEE Transactions on Power Systems*, vol. 27, no. 2, pp. 821-829, May 2012.
- [8] J. R. Silve and C. P. Arnold, "A simple improvement to sequential AC/DC power flow algorithms," *International Journal of Electrical Power & Energy Systems*, vol. 12, no. 3, pp. 291-221, Jul. 1990.
- [9] J. Beerten, S. Cole, and R. Belmans, "A sequential AC/DC power flow algorithm for networks containing multi-terminal VSC HVDC systems," in *Proceedings of IEEE PES General Meeting*, Providence, USA, Sept. 2010, pp. 1-7.
- [10] W. Feng, C. Yuan, Q. Shi *et al.*, "Using virtual buses and optimal multipliers to converge the sequential AC/DC power flow under high load cases," *Electric Power Systems Research*, vol. 177, pp. 1-9, Dec. 2019.
- [11] J. Wu and A. Bose, "Parallel solution of large sparse matrix equations and parallel power flow," *IEEE Transactions on Power Systems*, vol. 10, no. 3, pp. 1343-1349, Aug. 1995.
- [12] N. Garcia, "Parallel power flow solutions using a biconjugate gradient algorithm and a Newton method: a GPU-based approach," in *Proceedings of IEEE PES General Meeting*, Providence, USA, Sept. 2010, pp. 1-4.
- [13] C. Cheng, L. Bin, J. Shen *et al.*, "A modular parallelization framework for power flow transfer analysis of large-scale power systems," *Journal of Modern Power Systems and Clean Energy*, vol. 6, pp. 679-690, Jul. 2018.
- [14] A. Lumsdaine, D. Gregor, and B. Hendrickson, "Challenges in parallel graph processing," *Parallel Processing Letters*, vol. 17, no. 1, pp. 5-20, Mar. 2007.
- [15] T. Kajdanowicz, P. Kazienko, and W. Indyk, "Parallel processing of large graphs," *Future Generation Computer Systems*, vol. 32, pp. 324-337, Mar. 2014.
- [16] W. Feng, J. Wu, and C. Yuan, "A graph computation based sequential power flow calculation for large-scale AC/DC systems," in *Proceedings of 2019 IEEE PES General Meeting*, Atlanta, USA, Aug. 2019, pp. 1-5.
- [17] A. Goldberg and C. Harrelson, "Computing the shortest path: a search meets graph theory," in *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, Vancouver, Canada, Jan. 2005, pp. 156-165.
- [18] Y. Lu, J. Cheng, D. Yan *et al.*, "Large-scale distributed graph computing systems: an experimental evaluation," *Proceedings of the VLDB Endowment*, vol. 8, pp. 281-292, Nov. 2014.
- [19] R. Chen, J. Shi, Y. Chen *et al.*, "Powerlyra: differentiated graph computation and partitioning on skewed graphs," *Proceedings of the*

VLDB Endowment, vol. 5, Jan. 2019.

- [20] Q. Shi, C. Yuan, W. Feng *et al.*, "Enabling model-based LTI for large-scale power system security monitoring and enhancement with graph-computing-based power flow calculation," *IEEE Access*, vol. 7, pp. 167010-167018, Oct. 2019.
- [21] C. Yuan, Y. Zhou, G. Liu *et al.*, "Graph computing-based WLS fast decoupled state estimation," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2440-2451, May 2020.
- [22] C. Yuan, Y. Liu, W. Feng *et al.*, "Graph computing based distributed fast decoupled power flow analysis," in *Proceedings of 2019 IEEE PES General Meeting*, Atlanta, USA, Aug. 2019, pp. 1-5.
- [23] A. K. Srivastava, T. A. Ernster, R. Liu *et al.*, "Graph-theoretic algorithms for cyber-physical vulnerability analysis of power grid with incomplete information," *Journal of Modern Power Systems and Clean Energy*, vol. 6, pp. 887-899, Sept. 2018.
- [24] J. E. Gonzalez, Y. Low, H. Gu *et al.*, "PowerGraph: distributed graph-parallel computation on natural graphs," in *Proceedings of 10th USENIX Symposium on Operating Systems Design and Implementation*, Hollywood, USA, Oct. 2012, pp. 17-30.
- [25] A. Kyrola, G. Blelloch, and C. Guestrin, "GraphChi: large-scale graph computation on just a PC disk-based graph computation," in *Proceedings of 10th USENIX Symposium on Operating Systems Design and Implementation*, Hollywood, USA, Oct. 2012, pp. 31-46.
- [26] R. Xin, J. E. Gonzalez, M. J. Franklin *et al.*, "GraphX: a resilient distributed graph system on Spark," in *Proceedings of First International Workshop on Graph Data Management Experience and Systems*, New York, USA, Jun. 2013, pp. 1-6.
- [27] R. Dai, G. Liu, Z. Wang *et al.*, "A novel graph-based energy management system," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 1845-1853, May 2020.
- [28] L. G. Valiant, "A bridging model for parallel computation," *Communications of the ACM*, vol. 33, no. 8, pp. 103-111, Dec. 1990.
- [29] W. Feng, F. Li, C. Yuan *et al.*, "Graph computation based power flow for large-scale AC/DC system," in *Proceedings of 2018 International Conference on Power System Technology (POWERCON)*, Guangzhou, China, Sept. 2018, pp. 468-473.
- [30] M. Adibi. (1993, Nov.). IEEE 300-bus test case. [Online]. Available: http://labs.ece.uw.edu/pstca/pf300/pg_tca300bus.htm
- [31] Texas A&M University. (2018, Aug.). South Carolina 500-bus system. [Online]. Available: <https://electricgrids.engr.tamu.edu/electric-grid-test-cases/activsg500>

Wei Feng received the B.S. and M.S. degrees in Tsinghua University, Beijing, China, and the University of Chinese Academy of Sciences, Beijing, China, in 2012 and 2015, respectively. He received the Ph.D. degree in University of Tennessee, Knoxville, USA, in 2020. His research interests include high-performance computing, power flow analysis, and state estimation for hybrid AC/DC systems.

Chen Yuan received the B.S. degree in electrical engineering from Wuhan Uni-

versity, Wuhan, China, in 2012, and the Ph.D. degree in electrical and electronics engineering from Ohio State University, Columbus, USA, in 2016. He is a Senior Engineer with Global Energy Interconnection Research Institute North America, San Jose, USA. His current research interests include high-performance computing, power systems analysis, control systems, and energy management systems.

Qingxin Shi received the B.S. and M.Sc. degrees in Zhejiang University, Hangzhou, China, and University of Alberta, Edmonton, Canada, in 2011 and 2014, respectively. He received the Ph.D. degree in University of Tennessee, Knoxville, USA, in 2019, where he is working as a Research Assistant Professor. His research interests include demand response and distribution system resilience.

Renchang Dai received the Ph.D. degree in electrical engineering from Tsinghua University, Beijing, China, in 2001. He was with Amdocs, GE Global Research Center, Iowa State University, GE Energy Consulting, and GE/Alstom Grid Solution from 2002 to 2017. He is currently the Manager with Graph Computing and Grid Modernization Group, Global Energy Interconnection Research Institute North America, San Jose, USA. His research interests include smart grid, graph parallel computing, and AI and big data applications in power systems.

Guangyi Liu received the bachelor's and master's degrees from the Harbin Institute of Technology, Harbin, China, in 1984 and 1987, respectively, and the Ph.D. degree from China Electric Power Research Institute, Beijing, China, in 1990. He is the Chief Technology Officer with Global Energy Interconnection Research Institute North America, San Jose, USA. His research interests include energy management system and distribution management system, electricity market, smart grid, autonomous power system, active distribution network, and application of big data in power systems.

Zhiwei Wang received the B.S. and M.S. degrees in electrical engineering from Southeast University, Nanjing, China, in 1988 and 1991, respectively. He was the President of State Grid U.S. Representative Office, New York, USA, the President of State Grid Wuxi Electric Power Supply Company, and the Deputy Chief Engineer of State Grid Jiangsu Electric Power Company (JSEPC). He is the President of Global Energy Interconnection Research Institute North America. His research interests include power system operation and control, relay protection, power system planning, and wide-area measurement system (WAMS).

Fangxing Li, also known as Fran Li, received the B.S.E.E. and M.S.E.E. degrees from Southeast University, Nanjing, China, in 1994 and 1997, respectively, and the Ph.D. degree from Virginia Tech, Blacksburg, USA, in 2001. Currently, he is the James W. McConnell Professor in electrical engineering and the Campus Director of Center for Ultra-wide-area Resilient Electric Energy Transmission Networks (CURENT) at the University of Tennessee, Knoxville, USA. He is presently serving as the Editor-In-Chief of IEEE Open Access Journal of Power and Energy (OAJPE) and the Chair of IEEE/PES Power System Operation, Planning and Economics (PSOPE) Committee. His current research interests include renewable energy integration, demand response, distributed generation and microgrid, energy markets, and power system computing.