Splitting Strategies for Islanding Operation of Large-Scale Power Systems Using OBDD-Based Methods

Kai Sun, Da-Zhong Zheng, and Qiang Lu, *Fellow, IEEE*

Abstract—System splitting problem (SS problem) is to determine proper splitting points (or called splitting strategies) to split the entire interconnected transmission network into islands ensuring generation/load balance and satisfaction of transmission capacity constraints when islanding operation of system is unavoidable. For a large-scale power system, its SS problem is very complicated in general because a combinatorial explosion of strategy space happens. This paper mainly studies how to find proper splitting strategies of large-scale power systems using an OBDD-based three-phase method. Then, a time-based layered structure of the problem solving process is introduced to make this method more practical. Simulation results on IEEE 30- and 118-bus networks show that by this method, proper splitting strategies can be given quickly. Further analyses indicate that this method is effective for larger-scale power systems.

Index Terms—Graph theory, islanding operation, ordered binary decision diagrams, power system protection, splitting strategies, system splitting.

I. INTRODUCTION

FOR a power system, some serious disturbances may trigger growing oscillations, which lead to loss of synchronization between groups of generators and possibly blackouts. Normally system islanding may automatically happen after some transmission lines are tripped by local relays, but unbalanced electrical islands are often produced. Thus proper load shedding and generator tripping must be performed simultaneously. Fatally, if automatic system islanding only by local relays produces islands with excessive electrical unbalance (e.g., some islands are mainly made up of generators and, by contraries, the others almost have no generator but loads, then blackout of the entire system is almost inevitable). System splitting, also known as controlled system separation, is that dispatching center actively split the whole transmission network into two or several islands by tripping properly selected lines. After system splitting, the whole power system is under intentional islanding operation and each island of load and generation theoretically remains in balance. Thus, "although the power system is operating in an abnormal degraded state, customers are continuing to be served" [7]. The studies of historic blackouts or outages

[13]–[17] show the following two points: most blackouts arise in the incorrect operations of local relays that lead to islands with excessive electrical unbalance; if proper system splitting had been performed in time, many blackouts would have been avoided. So active and viable system splitting can efficiently avoid blackout of the entire power system and is ordinarily better than passive system islanding. Recent studies mainly aim at preventing system collapse and passive islanding [10], [15]–[18], detecting islanding and determining asynchronous groups of generators [4], [5], [8], dynamic analysis and stabilization control of the sub-system in each island (called "island system" below) [6]–[9], ameliorating functions and operation schemes of relays [9], [18], etc. But splitting strategies for how to split the whole system into islands, or in other words, which lines should be tripped, are seldom studied.

System splitting problem (SS problem) is just to determine proper splitting strategies that ensure synchronization of generators and satisfaction of "equality" and "inequality" constraints [12] in each island. It must be explained that after splitting only steady state stability of each island system is considered here, and in fact proper splitting strategies give necessary conditions of successful system splitting. That is based on the following two considerations: first, the decision of splitting should be given online (in an exceeding short time, perhaps several seconds), and hence, only the most important conditions against collapse of the entire system are considered here; second, any splitting strategies not satisfying these necessary conditions (even steady states are not stable) have no possibility to lead to successful system splitting; third, after system splitting according to a proper splitting strategy, available stabilization measures (e.g., island automatic-generation-control (AGC) function [7], etc., can make each island system operate normally. Moreover, the studies about how to find the splitting strategies satisfying those necessary conditions are lacking. In fact, the SS problem of a large-scale power system is generally very complicated because a combinatorial explosion of its strategy space is unavoidable. For example, for IEEE 118-bus network with 186 lines, there are altogether $2^{186} \approx 9.8 \times 10^{55}$ possible choices for system splitting. Its strategy space is too huge. Moreover, it is necessary to guarantee both speediness and correctness in determining the final splitting strategy to avoid collapse. A two-phase method based on ordered binary decision diagrams (OBDDs) [1], [2] to search for proper splitting strategies for not-too-large power systems is proposed in [3]. The method can search the whole strategy space using highly efficient OBDD-based algorithm in the first phase

and then give all proper splitting strategies in existence by power-flow analysis in the second phase. That demonstrates OBDD-based methods are feasible in solving SS problems.

This paper studies SS problems of large-scale power systems and uses an OBDD-based three-phase method to search for proper splitting strategies. In *Phase-1* of this method, a much simpler reduced network of the original power network is constructed by graph theory and the properties of SS problem itself. Then the verification algorithms based on OBDDs can efficiently narrow down the strategy space and can give enough splitting strategies satisfying "equality" constraints in *Phase-2*. In *Phase-3*, power-flow calculations are made to check if those strategies satisfy "inequality" constraints, and final optional proper splitting strategies will be given. Some concurrent technologies are applied in this method to speed up strategy search. Moreover, based this three-phase method, the assignment of offline and online tasks in solving SS problem is studied and a time-based layered structure of problem solving process is proposed, which can make this method more practical for large-scale power systems. Simulation results and further analyzes show that the method in this paper can efficiently find proper splitting strategies for large-scale power systems and make online strategy search realizable.

The rest of the paper is organized as follows. Section II introduces SS problem and OBDD representation. In Section III, how to solve SS problem by an OBDD-based three-phase method is introduced and then a time-based layered structure for the problem solving process is proposed. In Section IV, the method is applied to the IEEE 30- and 118-bus test network to demonstrate its performance; Section V provides some concluding remarks finally.

II. PRELIMINARIES

A. SS Problem and BP Problem

SS problem is to find the splitting strategies (called proper splitting strategies in this paper) that make the following three kinds of constraints satisfied after system splitting.

SSC: The whole power network is split into islands; generators in each island are synchronous approximately.
PBC: In each island, generation and load are balanced approximately. (*Equality* constraints)
RLC: Transmission lines and other transmission services must not be loaded above their transmission capacity limits (e.g., thermal capacity limits and steady state stability limits). (*Inequality* constraints).

Ensuring satisfaction of *SSC*, *PBC* and *RLC* is the most basic condition for successful system splitting. Typically, we only consider the case of splitting a power network into two islands in this paper. It is easy to extend the method in this paper to other cases. A subproblem of SS problem, BP problem, is given below, which has been proved NP-*complete*[3]. Hence, it follows that SS problem is NP-*hard*.

Balanced partition problem (BP problem):

Given an undirected, connected and node-weighted graph $G(V, E, W)$, two subsets V_{GA}, V_{GS} of V and a positive constant d, search for a subset of E (denoted by E_C) to split G into two connected sub-graphs $G_1(V_1, E_1, W_1)$ and

$G_2(V_2, E_2, W_2)$ such that $V_{GA} \subset V_1$, $V_{GS} \subset V_2$ (*SSC*) and the following constraint (*PBC*) are satisfied:

$$\left\| \sum_{v_i \in V_1} w_i \right\| \le d \text{ and } \left\| \sum_{v_j \in V_2} w_j \right\| \le d. \tag{1}$$

Here, $G(V, E, W)$ is the graph theory representation of a power network (called "graph-model" in [3]); w_i is the weight of v_i, which can be calculated by the following equation:

$$w_i = S_G^i - S_L^i \tag{2}$$

where S_G^i is the injected complex generator power and S_L^i is the complex load power at bus i; d, which is allowable power balance error limit, reflects the maximum acceptable unbalance of power in each island. In fact, some transmission lines do not react on system splitting but they may be cut off only to avoid violating *RLC*. Therefore, E_C contains a cut set (minimal edge set to separate a graph) of G, but it is not necessary to be a cut set. Moreover, in practical power network, local reactive power compensators can compensate reactive power for unbalance, and after system splitting real power balance and real power-flow in each island are more important. So we only consider real power balance here and (2) changes into (3). Thus, w_i is a real number

$$w_i = P_G^i - P_L^i. \tag{3}$$

B. OBDD Representation

An ordered binary decision diagram (OBDD) [1], [2] is a directed acyclic graph (DAG) representation of a Boolean expression. Generally, it is exponentially more compact than its corresponding truth table representation. In order to guarantee that equivalent Boolean expressions are uniquely represented, OBDDs are customarily requested to be reduced OBDDs [1]. There are many efficient algorithms to perform all kinds of logic operations on OBDD's. It is well-known that the problem *satisfiability of Boolean expressions* is NP-*complete* [19], but for the OBDD of a Boolean function $f(x_1, \ldots, x_n)$, denoted $D(f)$, the time complexity of checking its satisfiability is $O(n)$, where n is the number of variables. So once the OBDD of a Boolean function is built, its satisfiability will be verified in polynomial time. In addition, the choice of variable ordering of $x_1, \ldots, x_n$ can have a significant impact on the size of its OBDD. A person with some understanding of the problem domain can generally choose an appropriate variable ordering without difficulty to build an OBDD in acceptable size (generally, in polynomial size). Furthermore, if binary encoding is applied, an arbitrary integer variable m can be expressed by an OBDD vector, denoted $\mathrm{DV}(m)$, whose each OBDD element represents one binary bit of that integer variable. Consequently, any algebraic expression only including integers and integer variables can be represented by OBDD's. In practice, many BDD software packages [26]–[30] provide algorithms of OBDD vectors.

Since BP problem is *NP*-complete and contains a large number of decision variables corresponding to transmission lines in a power network, so it is reasonable to believe that a good representation (or data structure) for BP problem can

effectively improve solving efficiency. OBDD is just one of such good representations, whose powers have been shown by large industry examples [2], [21]–[24]. One of the most powerful applications of OBDDs is symbolic model checking [24] used to formally verify digital circuits and other finite state systems. All these practical examples show that, for symbolic verification problems or satisfiability checking problems, whose variables generally belong to infinite sets, especially integer or Boolean variables, OBDD presentation is very efficient. We summarize below the main advantages of OBDD-based methods to SS problem or BP problem

1) BP problem is a typical satisfiability checking problem and can be easily represented by OBDDs.
2) OBDD-based searching algorithms can perform the search of whole searching space; hence, they are essentially different from many stochastic searching methods [e.g., genetic algorithm (GA), simulated annealing, etc., which may fail to find only one solution]. However, once the OBDD of a resoluble problem is built, all of its solutions will be found in quite short time. That is a very important advantage to SS problem.
3) OBDD representation provides a systematic way to explore the structural information of a complex problem. A typical illustration is that selecting reasonable variable ordering can significantly reduce the combinatorial explosion of searching space. So the structural characters of BP problem (e.g., the topological characteristics of power network), should be adequately applied to select a reasonable variable ordering.
4) Building OBDDs by stages is supported. In order to build the OBDD of a complex real-time decision problem, we firstly partition the original problem into several subproblems and then build respective OBDDs in different time. For example, OBDDs of some subproblems are changeless; hence, they can be built offline. Thus online tasks are effectively reduced. Finally, all OBDDs built in different times can be quickly combined into one OBDD by APPLY operations of OBDDs [1].

III. SOLVING SYSTEM SPLITTING PROBLEM

A. An OBDD-Based Three-Phase Method for SS Problem

In order to solve SS problems for large-scale power systems, an OBDD-based three-phase method is introduced in this section, which is adapted from the two-phase method in [3]. Its three phases are followed

Phase-1: Initialize parameters (e.g., d); reduce the original complex power network by graph theory and properties of power network itself.

Phase-2: Solve BP problem of the reduced network, or in other words, search for all strategies satisfying *SSC* and *PBC* by OBDD-based search algorithms. If BP problem has no solution, then go back to *Phase-1* (change initial parameters, relax some constraints and modify the reduced network).

Phase-3: Check *RLC* of these strategies by power-flow calculations. If some strategies satisfy *RLC*, then give them as
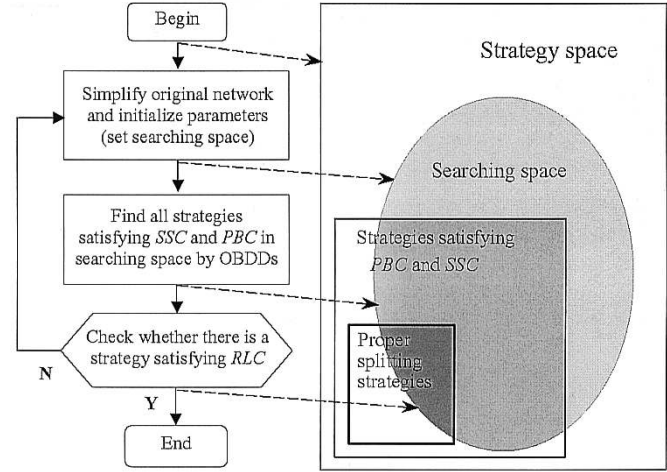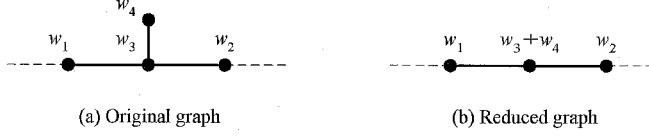

Fig. 1. Flowchart of the three-phase method and the reduction process of searching space.

optional proper splitting strategies. Otherwise go back to *Phase-1*.

The main idea of this method is reducing strategy space by phases unto a set including proper splitting strategies. The aim of *Phase-1* is to simplify the original complex network to a more manageable network; *Phase-2* and *Phase-3*, which are respectively adapted from the two phases of that two-phase method, become more efficient since some concurrent technologies are applied in solving BP problem and power-flow calculations. The flowchart of this three-phase method and the reduction process of strategy space are shown in Fig. 1. In addition, it must be pointed out that because of the hugeness of strategy space of a large-scale power network, there are, in general, enough proper splitting strategies satisfying *SSC*, *PBC* and *RLC*. However, when a power system needs system splitting imminently, only one feasible and safe splitting strategy is enough. So it is not necessary to find all proper splitting strategies in strategy space. In fact, in order to accelerate the searching process, the action of *Phase-1* is to reduce searching range of strategy space to a smaller region called "searching space" (the ellipse in Fig. 1), which discards some proper splitting strategies but still contains enough proper splitting strategies. Of course, for a small-scale power system, *Phase-1* is not necessary and can be omitted. However, for a power system much larger than IEEE 30-bus system, the function of *Phase-1* is very remarkable. Thus, *Phase-2* and *Phase-3* are both carried out in searching space and, as shown in Fig. 1, their results are intersections of the ellipse and respective rectangles. How to find proper splitting strategies by this three-phase method and the function of each phase will be detailed in the following Sections III-C and D.

B. Phase-1: Simplifying Original Power Network

The complexity of BP problem highly depends on the size of power network especially the number of edges in E, which is also equal to the number of decision variables in system splitting. Therefore, we should attempt to simplify original power network by graph theory and properties of power network itself. In the rest of this paper, we respectively use $G(V, E, W)$ and
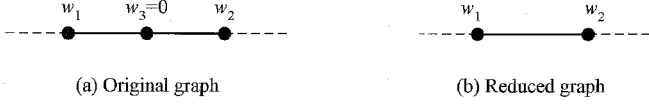
Fig. 2.   Node v_3 can be removed.

Fig. 3.   Nodes v_3 and v_4 can be merged.

Fig. 4.   Edge e_{12} can be cut off.

Fig. 5.   An example of "edges can be cut off."

(a)

(b)

Fig. 6.   IEEE 30-bus network.

$G^r(V^r, E^r, W^r)$ to denote the graph-models of original power network and its reduced power network. E_C and E_C^r are just the solutions of their BP problems, respectively. Then we present two kinds of effective simplification methods called *SM-1* and *SM-2*.

1) SM-1: Reducing Irrelevant Nodes and Edges: There are three kinds of typical cases that can be simplified in most power networks, which will be given below by graph-model. In the following three figures, the weight of each node is given nearby, and the meanings of broken lines will be explained, respectively.

   a) *Nodes can be removed:* In Fig. 2, the nodes connected with broken lines may be terminals or connected with other subgraphs by arbitrary number of edges. As shown in Fig. 2(a), since $w_3 = 0$, it is generally unimportant to differentiate which of e_{13} and e_{23} is cut off in system splitting. So the nodes like v_3 may be removed as shown in Fig. 2(b) (in fact, combined with v_1 or v_2). Thus, the elements of E and V all subtract 1.

   b) *Two nodes can be merged:* Broken lines in Fig. 3 denote the same as *a)*. In this paper, we do not consider the case of islanding only one node from network (called plant separation [4], [11]), [e.g., islanding v_4 in Fig. 3(a)]. Thus, v_3 and v_4 may be merged into a new node with weight $w_3 + w_4$ as shown in Fig. 3(b), and the elements of E and V all subtract 1.

   c) *Edges can be cut off:* In Fig. 4, the thin broken lines connected with v_1 or v_2 denote the same as *a)* but they cannot both denote 0 edge. G_S is an isolated subgraph (perhaps has no node). The thick broken line connecting v_1 (or v_2) with G_S denotes an arbitrary nonzero number of edges. The simplest case of Fig. 4(a) is that v_1 and v_2 are directly connected by two edges. If (4) is satisfied, it is impossible to island G_S. Thus, it is obvious that cutting off e_{12} is a necessary condition of separating v_1 from v_2. So e_{12} is redundant in deciding whether v_1 and v_2 are connected and may be removed before solving BP problem. For instance, in Fig. 5, e_{56}, e_{78}, e_{79} and either of e_{13} and e_{23}
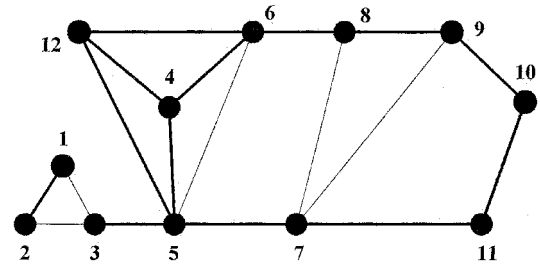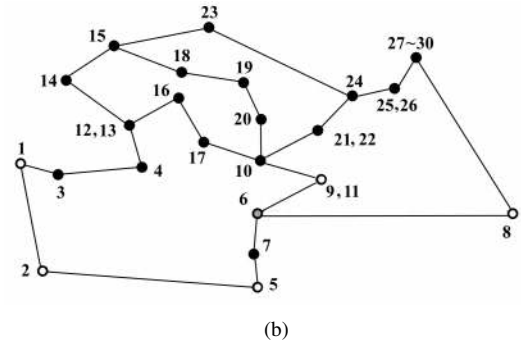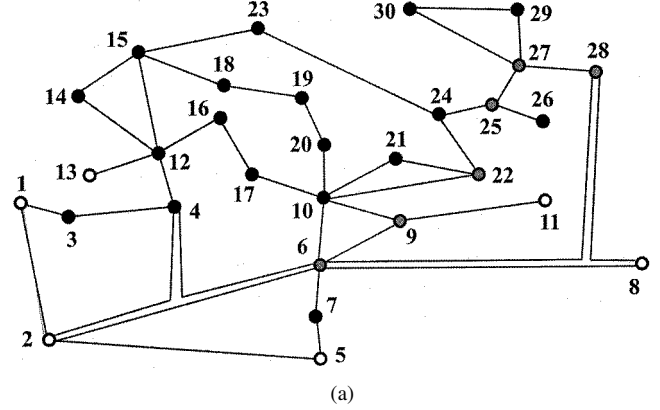
may be cut off after simplification if their (4)'s are satisfied

$$\left\| \sum_{v_i \in G_S} w_i \right\| > d. \tag{4}$$

   In practice, the three measures of *SM-1* should be applied repeatedly till the above three cases are not existent any longer. Assume that in graph-model black dots denote load nodes (only connected with loads), white dots denote generator nodes (connected with at least one generator and perhaps together with associated loads) and gray dots denote the nodes with weight 0. As shown in Fig. 6, if we simplify IEEE 30-bus network by *SM-1*, its G^r is given in Fig. 6(b) [the number of nodes is reduced from 30 to 23 and the number of edges is reduced from 41 to 26].

2) SM-2: Combining Nodes by Area: *SM-1* is often not efficient enough to simplify a large-scale power network for *Phase-2*. So *SM-2* is needful before performing *SM-1*. This simplification method is proposed based the following consideration: the lines tripped in system separation belong to transmission network not distribution network and, generally,

have higher voltage ratings, or in other words, a great number of lines absolutely should not be considered in system splitting. For example, it is unrealistic to consider whether the lines of a school should be cut off in the city's power system separation. For this reason, we divide all nodes of G into N_e areas by their voltage ratings or their own geographic areas. For example, if we split the whole power system of a country, these areas can be decided by provinces or regions. Treat adjoining nodes in an area as a whole and combine them to make an equivalent new node, whose weight is equal to the sum of the weights of all nodes in this area. The new node is a generator node if this area includes at least one generator node. Only hold one edge for every two connected areas. Then, a N_e-node new graph is produced.

In the partition of areas, if we request that real-power load of every pure load area (without generators) does not exceed a reasonable upper limit, denoted P_{Area}, the following approximate approach is useful to estimate P_{Area} and N_e. Obviously, P_{Area} is directly connected with d and d should make the frequency offset of each island system no more than the prescriptive frequency offset limit Δf. The integrality of each island system and the reliability of power supply are most important. So comparing with normal power systems, a little larger Δf (e.g., 1–2 Hz) in each island system is allowable. Assume that the ability of real-power generation of each island system is requested larger than P_{Island} and rating frequency $f_0 = 60$ Hz. From speed-power curve, the *regulation constant* [12], denoted R, of every island system satisfies

$$R < \sigma \cdot \frac{f_0}{P_{\mathrm{Island}}} \text{ (Hz/MW)} \tag{5}$$

where constant $\sigma = 2$–5%. Approximately we have

$$d \le d_{\max} \doteq \frac{\Delta f \cdot P_{\mathrm{Island}}}{\sigma \cdot f_0} < \frac{\Delta f}{R} \tag{6}$$

where $d_{\max}$ is the upper limit of d. For the sake of large enough searching space after *Phase-1*, P_{Area} should not much larger then $d_{\max}$. For example, we may prescript $P_{\mathrm{Area}} < d_{\max}$. Finally, N_e can be determined by P_{Area}.

In addition, *SM-1* should be performed after *SM-2*. It should be noted that the nodes like v_4 in Fig. 3 should not appear after *SM-2* because they will certainly be merged into adjacent nodes after performing *SM-1* and possibly the weights of some merged nodes will exceed P_{Area}. Finally, a reduced network G^r is produced. We assume that it has N nodes and M edges.

The significance of *Phase-1* must be emphasized, however large the original power network is, as long as it is simplified to a simple enough network G^r in *Phase-1*, *Phase-2* will only face G^r and solve its BP problem. So N and M should not be too large. For example, by the computer resources similar to that of the simulations below, $N < 30$ and $M < 30$ are recommended. Moreover, we must point out: first, the performance of our method is mainly determined by the magnitudes of final N and M, and in which way all nodes in G are divided into N_e areas is relatively not important; second, how many nodes one area can contain is not confined. In fact, we may regard a city's or a province's power system as one area, which generally contains thousands of nodes. In conclusion, with the aid of *Phase-1*,
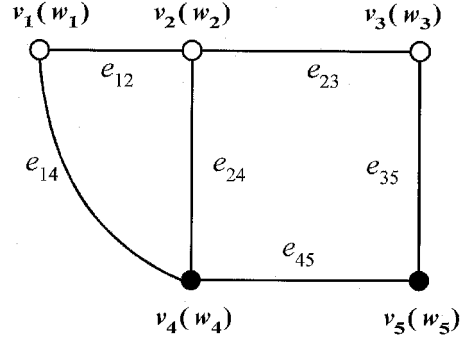

Fig. 7.   Graph-model of a five-bus power system.

we can highly simplify the SS problem of a large-scale power system and make it possible to find proper splitting strategies.

C. Phase-2: Solving BP Problem

A representation of BP problem in propositions can be found in [3]. For the convenience of building the OBDD of BP problem, Boolean function representation is applied here. We first present *SSC* and *PBC* in Boolean functions. Then, their OBDDs are built, respectively. Finally, the two OBDDs are combined to make the OBDD of BP problem by APPLY operation "AND" [1]

$$A_G = \begin{bmatrix} 0 & b_{12} & 0 & b_{14} & 0 \\ b_{12} & 0 & b_{23} & b_{24} & 0 \\ 0 & b_{23} & 0 & 0 & b_{35} \\ b_{14} & b_{24} & 0 & 0 & b_{45} \\ 0 & 0 & b_{35} & b_{45} & 0 \end{bmatrix}. \tag{7}$$

We use A_G to denote the adjacency matrix of the N-node reduced network $G^r(V^r, E^r, W^r)$ given by *Phase-1*. The elements $(A_G)_{ij}$ and $(A_G)_{ji}$ of A_G are the same Boolean variable, denoted b_{ij} (here assume that $i < j$), if there is a edge $e_{ij}^r \in E_C^r$; otherwise $(A_G)_{ij} = (A_G)_{ji} = 0$. For example, the A_G of a five-bus power system shown in Fig. 7 is given as (7). In fact, to decide $b_{ij} = 0$ (or 1) is just to decide whether $e_{ij}^r \in E_C^r$. Consequently, to solve the BP problem of G^r is to decide all M Boolean variables b_{ij}. If we, respectively, use "AND" and "OR" as Boolean multiplication and addition operators, denoted "$\otimes$" (often omitted) and "$\oplus$," then it is easy to draw the following conclusions from Boolean matrix theory [20].

1) Element $(A_G^k)_{ij}$ is a polynomial with "$\otimes$" and "$\oplus$," if there is a path of length $\le k$ from v_i^r to v_j^r in G^r; otherwise, $(A_G^k)_{ij} = 0$. Dually, all paths of length $\le k$ from v_i^r to v_j^r in G^r determine and are determined by $(A_G^k)_{ij}$.

2) If we define

$$A_G^* \overset{\mathrm{def}}{=} I \oplus A_G^1 \oplus A_G^2 \oplus \cdots \oplus A_G^L \tag{8}$$

where L is the longest path in G and I is identity matrix (1 on the diagonal and 0 elsewhere), then A_G^* can determine the connection of arbitrary two nodes after arbitrary unknown graph partition.

For instance, considering of the above five-bus power system and all the paths from v_1 to v_4 in Fig. 7, we have

$$(A_G^*)_{14} = b_{14} \oplus b_{12}b_{24} \oplus b_{12}b_{23}b_{35}b_{45} \tag{9}$$

where b_{14}, $b_{12}b_{24}$, and $b_{12}b_{23}b_{35}b_{45}$, respectively, correspond with paths e_{14}, $e_{12}e_{24}$, and $e_{12}e_{23}e_{35}e_{45}$. If the system is split into two islands by cutting off e_{23} and e_{45}, then path $e_{12}e_{23}e_{35}e_{45}$ will be interdicted. It follows that $(A_G^*)_{14} = b_{14} \oplus b_{12}b_{24}$, which can also be obtained by setting $b_{23} = 0$ and $b_{45} = 0$ in (9). For the convenience of expression, we define the following three sets of serial numbers:

$$I_A = \{i | v_i^r \in V_{GA}^r\}, \quad I_S = \{j | v_j^r \in V_{GS}^r\} \text{ and}$$
$$I_G = I_A \bigcup I_S \tag{10}$$

I_G contains the serial numbers of all generator nodes. Let N_G denote the total number of generator nodes. Select arbitrary two elements $i_A \in I_A$ and $i_S \in I_S$. Then, *SSC* can be expressed as

$$\text{SSC} = \prod_{\forall i \in I_A, \forall j \in I_S}^{\otimes} \overline{(A_G^*)_{ij}} \otimes \prod_{\forall i \in I_A}^{\otimes} (A_G^*)_{i,i_A}$$
$$\otimes \prod_{\forall j \in I_S}^{\otimes} (A_G^*)_{j,i_S} \otimes \prod_{\forall k \notin I_G}^{\otimes} \left[(A_G^*)_{k,i_A} \bar{\oplus} (A_G^*)_{k,i_S} \right] \tag{11}$$

where "$\bar{\oplus}$" denotes "EXCLUSION-OR" operator. It is easy to know that, in (11), the first multiplier factor is redundant since the final multiplier factor exists. So *SSC* can be reduced to

$$\text{SSC} = \prod_{\forall i \in I_A}^{\otimes} (A_G^*)_{i,i_A} \otimes \prod_{\forall j \in I_S} \otimes (A_G^*)_{j,i_S}$$
$$\otimes \prod_{\forall k \notin I_G}^{\otimes} \left[(A_G^*)_{k,i_A} \bar{\oplus} (A_G^*)_{k,i_S} \right]. \tag{12}$$

Obviously, *SSC* is a Boolean function of all b_{ij}, which correspond to all edges in G^r ($\forall e_{ij}^r \in E^r$). Given a proposition P, we define

$$\langle P \rangle = \begin{cases} 1, & \text{if } P \text{ is true} \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Then, *PBC* can be expressed as

$$\text{PBC} = \prod_{i=1\cdots N}^{\otimes} \langle |(A_G^*)_{i*} \cdot W| \leq d \rangle \tag{14}$$

where $(A_G^*)_{i*}$ is the i-th row of A_G^* and $W = [w_1^r, \ldots, w_N^r]^T$ is the weight vector of N nodes. Assuming that *SSC* is satisfied, *PBC* can be simplified as

$$\text{PBC} = \left\langle \left| \sum_{k \notin I_G} (A_G^*)_{i_S,k} \cdot w_k^r + \sum_{l \in I_S} w_l^r \right| \leq d \right\rangle$$
$$\otimes \left\langle \left| \sum_{g \notin I_G} (A_G^*)_{i_A,g} \cdot w_g^r + \sum_{h \in I_S} w_h^r \right| \leq d \right\rangle. \tag{15}$$

In fact we only need calculate the i_A-th and i_S-th rows (or columns) of elements in A_G^* from (12) and (15). The Boolean function representation of *PBC* can be obtained by the following two approaches

1) Solve the following equation group about Boolean variables $(A_G^*)_{ij}$ ($i = i_A$ or i_S, $j \notin I_G$). Write its solution in the form of Boolean expression with "$\otimes$" and "$\oplus$." Then, the Boolean function representation of *PBC* can be obtained [see equation (16) at the bottom of the page].

2) Binary encoding is applied. First, we represent $w_1^r, \ldots, w_N^r$ and d by binary codes. Thus they must be replaced by integers $\lceil \lambda \cdot w_1^r \rceil, \ldots, \lceil \lambda \cdot w_N^r \rceil$ and $\lceil \lambda \cdot d \rceil$ together, where "$\lceil \ \rceil$" (called "ceil operation") obtains the nearest integer larger than a real number and λ is a big enough integer to decrease rounding errors. In order to reduce the binary bits used in encoding, denoted n_b, we apply modular arithmetic to represent all integers (including negative integers) by $0 \sim 2^{n_b} - 1$. Thus, any integer x is replaced by $x(\text{mod} 2^{n_b})$. Since I_A must be separated from I_S finally, n_b can be selected by (17)

$$n_b \geq \left\lceil \log_2 \left[\lambda \cdot \max \left(\sum_{i \in I_A} w_i^r, \sum_{j \in I_S} w_j^r \right) \right] \right\rceil. \tag{17}$$

Second, the integer operations in (14) are translated into equivalent logic operations. For example, consider a common example of addition operations: $X + Y = Z$, where X, Y and Z are all integers less than 8. Assuming that the 3-b binary encoding strings of X, Y and Z are $[x_3 x_2 x_1]$, $[y_3 y_2 y_1]$, and $[z_3 z_2 z_1]$, we have

$$[x_3 x_2 x_1] + [y_3 y_2 y_1] = [z_3 z_2 z_1]. \tag{18}$$

There is the following relationship:

$$z_i = (x_i \bar{\oplus} y_i) \bar{\oplus} c_{i-1}, \text{ where}$$
$$c_{i-1} = (x_{i-1} \otimes y_{i-1}) \oplus [c_{i-2} \otimes (x_{i-1} \oplus y_{i-1})] \text{ and}$$
$$c_0 = 0. \tag{19}$$

Then the Boolean function representation of *PBC* can be obtained easily. In fact, many BDD software packages support OBDD vector representation. They can encode any integer expressions and build their OBDD's by similar approaches. Given n_b and a variable ordering, most of them can directly build $D(PBC)$ from (14) or (15).

As an illustration, consider the five-bus power system shown in Fig. 7. Assume that $w_1 = 0.2$, $w_2 = 0.3$, $w_3 = 0.4$, $w_4 = -0.5$, $w_5 = -0.4$, $I_S = \{1, 2\}$, $I_A = \{3\}$, and $d = 0.1$.

$$\begin{cases} \left| \sum_{k \notin I_G} (A_G^*)_{i_S,k} \cdot w_k^r + \sum_{l \in I_S} w_l^r \right| \leq d \\ \left| \sum_{g \notin I_G} (A_G^*)_{i_A,g} \cdot w_g^r + \sum_{h \in I_S} w_h^r \right| \leq d \\ (A_G^*)_{ij} = 0 \text{ or } 1 \qquad\qquad (i = i_A \text{ or } i_S, \ j \notin I_G). \end{cases} \tag{16}$$

Phase-1 is omitted. Selecting $i_A = 2$ and $i_S = 3$, we have (20) from (12),

$$\text{SSC} = (A_G^*)_{12} \otimes [(A_G^*)_{24} \bar{\oplus} (A_G^*)_{34}]$$
$$\otimes [(A_G^*)_{25} \bar{\oplus} (A_G^*)_{35}] \quad (20)$$

where $(A_G^*)_{12} = b_{12} \oplus b_{14}b_{24} \oplus b_{14}b_{45}b_{35}b_{23}$, $(A_G^*)_{24} = b_{24} \oplus b_{12}b_{14} \oplus b_{23}b_{35}b_{45}$, $(A_G^*)_{34} = b_{23}b_{24} \oplus b_{35}b_{45} \oplus b_{12}b_{14}b_{23}$, $(A_G^*)_{25} = b_{23}b_{35} \oplus b_{24}b_{45} \oplus b_{12}b_{14}b_{45}$ and $(A_G^*)_{35} = b_{35} \oplus b_{23}b_{24}b_{45} \oplus b_{12}b_{23}b_{14}b_{45}$. From (15), we have

$$\text{PBC} = [|w_1 + w_2 + (A_G^*)_{24} \cdot w_4 + (A_G^*)_{25} \cdot w_5| \leq d]$$
$$\otimes [|w_3 + (A_G^*)_{34} \cdot w_4 + (A_G^*)_{35} \cdot w_5| \leq d]. \quad (21)$$

First, apply the first approach and solve equation group (22)

$$\begin{cases} |0.2 + 0.3 - 0.5 \cdot (A_G^*)_{24} - 0.4 \cdot (A_G^*)_{25}| \leq 0.1 \\ |0.4 - 0.5 \cdot (A_G^*)_{34} - 0.4 \cdot (A_G^*)_{35}| \leq 0.1 \\ (A_G^*)_{24}, (A_G^*)_{25}, (A_G^*)_{34} \text{ and } (A_G^*)_{35} = 0 \text{ or } 1. \end{cases} \quad (22)$$

It follows that the only solution is $(A_G^*)_{24} = 1$, $(A_G^*)_{25} = 0$, $(A_G^*)_{34} = 0$, and $(A_G^*)_{35} = 1$. So we have

$$\text{PBC} = (A_G^*)_{24} \otimes (A_G^*)_{35} \otimes \overline{(A_G^*)_{25}} \otimes \overline{(A_G^*)_{34}}. \quad (23)$$

Then, apply the second approach. If we select $\lambda = 10$, then $n_b \geq 3$ from (17). Let $n_b = 3$. Applying modular arithmetic, we have

$$\lceil \lambda \cdot w_1 \rceil (\text{mod}8) = 2, \quad \lceil \lambda \cdot w_2 \rceil (\text{mod}8) = 3,$$
$$\lceil \lambda \cdot w_3 \rceil (\text{mod}8) = 4, \quad \lceil \lambda \cdot w_4 \rceil (\text{mod}8) = 3,$$
$$\lceil \lambda \cdot w_5 \rceil (\text{mod}8) = 4, \quad \lceil \lambda \cdot d \rceil (\text{mod}8) = 1.$$

Thus, we can rewrite *PBC* as

$$\text{PBC} = \langle 2 + 3 + 3 \cdot (A_G^*)_{24} + 4 \cdot (A_G^*)_{25} = 0 \rangle$$
$$\otimes \langle 4 + 3 \cdot (A_G^*)_{34} + 4 \cdot (A_G^*)_{35} = 0 \rangle$$
$$= \langle 3 \cdot (A_G)_{24} + 4 \cdot (A_G^*)_{25} = 3 \rangle$$
$$\otimes \langle 3 \cdot (A_G^*)_{34} + 4 \cdot (A_G^*)_{35} = 4 \rangle$$
$$= \langle [0, (A_G)_{24}, (A_G)_{24}] + [(A_G^*)_{25}, 0, 0] = [011] \rangle$$
$$\otimes \langle [0, (A_G^*)_{34}, (A_G^*)_{34}] + [(A_G^*)_{35}, 0, 0] = [100] \rangle. \quad (24)$$

From (24) and (19), we can obtain the same result as (23). For most detailed informations about binary encoding, see [2].

In order to yield compact OBDDs in representing *SSC* and *PBC*, we give the following approach to decide the variable ordering of b_{ij}, which can be referred to as a breadth-first ordering.

1) Renumber all nodes
   a) Remove the original numbers of all nodes. Number all generator nodes from 1 to N_G by their locations in network. Let $k = 1$ and $O = N_G + 1$.
   b) If there is an unnumbered node connected with v_k^r, then number it O and let $O = O + 1$; else $k = k + 1$. If $k > N$, then turn to 2); else perform b) again.
2) Define $\text{EN}(b_{ij}) = i \cdot N + j(i < j)$ and order all b_{ij} according to the increase ordering of their $\text{EN}(b_{ij})$.

Variable orderings selected by this approach consider the structural information of G^r, so they are more efficient than
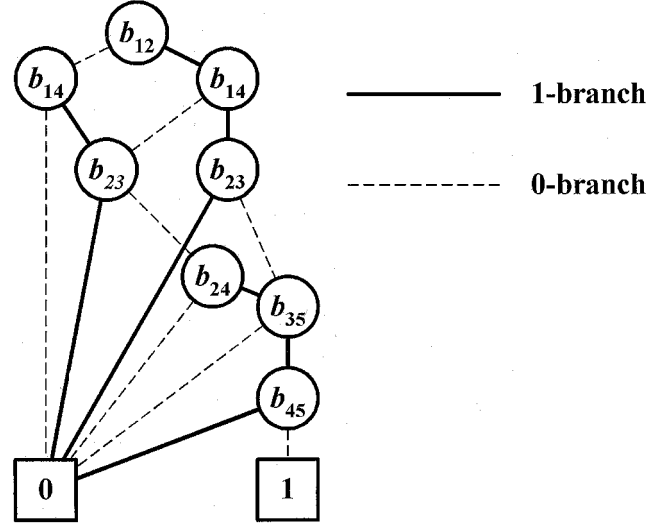

Fig. 8. *D(BP)* of five-bus power system.

random orderings generally. Then, $D(SSC)$ and $D(PBC)$ can be easily built by BDD software packages. Finally, the APPLY operation "AND" can quickly merge them into *D(BP)*

$$D(BP) = \text{APPLY}(\text{AND}, D(SSC), D(PBC)). \quad (25)$$

Then, a solution of BP problem (E_C^r) can be obtained by "*satisfy-one*" procedure. We even can obtain all solutions or the number of solutions by "*satisfy-all*" and "*satisfy-count*" procedures. They can be implemented by BDD software packages and their original algorithms can be found in [1]. For instance, considering that five-bus power system and selecting variable ordering $b_{12} < b_{14} < b_{23} < b_{24} < b_{35} < b_{45}$, we can construct its *D(BP)*, as shown in Fig. 8.

In *D(BP)* with M variables, any path that passes m different variable nodes from root node to "1"-node corresponds to 2^{M-m} solutions of BP problem. For Fig. 8, all four solutions (E_C's) can be obtained from corresponding paths $b_{12}b_{14}\bar{b}_{23} b_{35}\bar{b}_{45} \rightarrow$, $\{e_{23}, e_{45}\}$, and $\{e_{23}, e_{24}, e_{45}\}$; $b_{12}\bar{b}_{14}\bar{b}_{23}b_{24}b_{35}\bar{b}_{45} \rightarrow \{e_{14}, e_{23}, e_{45}\}$; $\bar{b}_{12}b_{14}\bar{b}_{23}b_{24}b_{35}\bar{b}_{45} \rightarrow \{e_{12}, e_{23}, e_{45}\}$.

It must be noticed that the G^r is only used in *Phase-2*. From every E_C^r, corresponding E_C's (generally, more than one) must be yielded before *Phase-3* starts. How to obtain them is discussed below. It is obvious that E_C^r must contain a cut set of G. Thus, the edges in G whose corresponding edges in G^r belong to that cut set must belong to E_C. For example, in Fig. 2, if $e_{12}^r \in E_C^r$, either of e_{13} and e_{23} belongs to E_C. In practical terms, some transmission lines not reacting on system splitting may still be cut off only to avoid violating *RLC*. Accordingly, any edges may belong to E_C as long as cutting off them will not create other isolated subgraphs except G_1 and G_2. Assume that E_C^r splits G^r into $G_1^r(V_1^r, E_1^r, W_1^r)$ and $G_2^r(V_2^r, E_2^r, W_2^r)$. For convenience, we apply the following two guidelines in the simulations below to get only one E_C from every E_C^r: if an edge belongs to E_C^r, then its corresponding edges in G belong to E_C; in G, an edge e_{kl} belongs to E_C if the corresponding nodes of v_k and v_l in G^r, respectively, belong to V_1^r and V_2^r; the other edges does not belong to E_C.

D. Phase-3: Power-Flow Calculations

In this phase, for every E_C given by *Phase-2*, power-flow calculation is carried out on $G - E_C$ (all island systems) and then the power in each transmission line is obtained. Typically, we only consider the transmission line capacity limits in *RLC*. In order to speed up power-flow calculations to give proper splitting strategies in time and avoid passive collapse of system, high accuracy of the calculation is not necessary in *Phase-3*. Moreover, since we only consider real power balance in system splitting, a fast power-flow calculation method *DC power-flow* is applied here. Given real powers of all buses, which are equal to $w_i (i = 1, \ldots, n)$ in G and can be obtained from load forecasting data, we can calculate the real power in each line (denoted P_{ij}) by solving a group of linear equations. Normally, its calculation errors are in 3%–10%. In order to ensure satisfaction of *RLC*, those errors must be considered. An efficient measure is to strengthen the steady security limits of transmission lines (denoted PSL_{ij}) to ensure certain margin of safety. For example, we may set a safety coefficient $\alpha \in [0.7, 0.9]$. Thus, an E_C given by *Phase-3* is a proper splitting strategy if, for every e_{ij}, $P_{ij} < \alpha \cdot \mathrm{PSL}_{ij}$. Furthermore, in order to shorten the time for *Phase-3*, concurrent calculations should be applied in power-flow calculations: *RLC* of different E_C's can be checked concurrently by many computers; on the other hand, P_{ij} in different island systems can also be calculated by concurrent computers. For large-scale power network, we even may perform *Phase-3* only on the backbone of power system and only check *RLC* for the transmission lines with high voltage ratings.

E. Partitioning the Tasks in Solving SS Problem

In practical operation, in order to shorten online searching time for proper splitting strategies, it is rational to perform some tasks before system splitting is required. For example, we can perform some tasks periodically as preparations for potential system splitting. So we partition all tasks in three phases into the following three kinds.

1) offline tasks: they are performed infrequently (from once in several days to once in several months);
2) periodic tasks: they are periodically performed only at intervals of T_D (the period of very short-term load forecasting);
3) online tasks: they may begin at any moment and must be completed in several seconds.

In general, the topological characteristic of G is changeless. So *SM-1* can be performed offline. As long as every two generators that have possibilities to become asynchronous are divided into different areas, *SM-2* can also be performed offline. It follows that constructing G^r and $D((A_G^*)_{ij})$ $(i = 1, \ldots, N_G, j = 1, \ldots, N)$ are offline tasks. Thus BP problem will be solved directly as soon as system splitting is required. $w_1^r, \ldots, w_N^r$ are obtained from very short-term load forecasting, whose period T_D is about 10 min–1 h. So the OBDD vectors of $w_1^r, \ldots, w_N^r$ may be built at intervals of T_D (once very short-term load forecasting is completed). In order to build *D(PBC)* only at intervals of T_D too, we have to use (14) not (15) since I_A and I_S are gen-
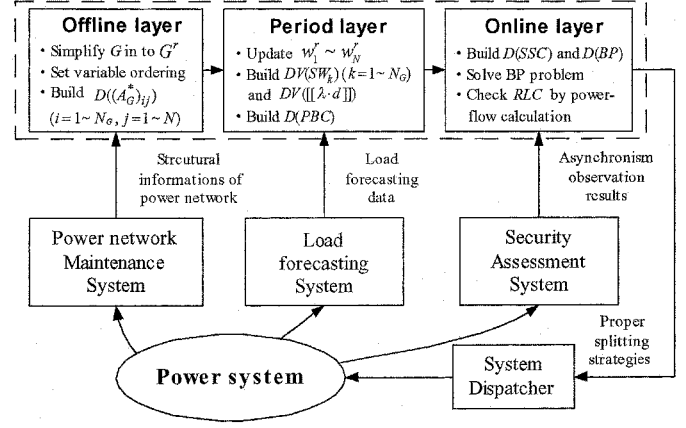

Fig. 9. Time-based layered structure for solving SS problem.

erally impossible to know. Moreover, (17) must be replaced by (26)

$$n_b \geq \left\lceil \log_2 \left[\lambda \cdot \sum_{i \in I_G} \max(w_i^r, 0) \right] \right\rceil. \qquad (26)$$

From (14), for every generator node v_k^r, we must calculate $\sum_{i=1,\ldots,N} (A_G^*)_{ki} \cdot \lceil \lambda \cdot w_i^r \rceil$, denoted SW_k, and build its OBDD vector $\mathrm{DV}(\mathrm{SW}_k)$. In practical operation, $\mathrm{DV}(\mathrm{SW}_1)$–$\mathrm{DV}(\mathrm{SW}_{N_G})$ can concurrently be built by N_G computers. Hence, the total time used in building *D(PBC)* is approximately equal to the longest of times for each $\mathrm{DV}(\mathrm{SW}_k)$ plus the time for forming *D(PBC)* by $\mathrm{DV}(\mathrm{SW}_1)$–$\mathrm{DV}(\mathrm{SW}_{N_G})$. Although building *D(PBC)* in this way may cost more time, it need not be performed online and becomes periodic tasks. Finally, building *D(SSC)* and *D(BP)*, solving BP problem and checking *RLC* by power-flow calculations are online tasks.

Based on the above partition of tasks, we proposed a time-based layered structure of problem solving process, which makes this three-phase method more practical and aims at realizing real-time decision making and designing decision support systems for system splitting. As shown in Fig. 9, it has three time layers, respectively, handling the above three kinds of tasks: offline layer, periodic layer, and online layer.

IV. SIMULATION RESULTS

The adaptations of standard IEEE 30-bus and IEEE 118-bus networks are used to demonstrate the performance of our method. We select BuDDy package (v2.0) [25], which supports all standard OBDD operations and especially many highly efficient OBDD vector operations, to program by C++ language on PC (Pentium IV-1.4G CPU and 256M DDRAM) according to the above time-based layered structure.

A. Simulation Results of IEEE 30-Bus Network

G and G^r are shown in Fig. 6. *SM-2* is not used. $N = 23$, $M = 26$, and $N_G = 6$. Here we only solve its BP problem. The generation (P_G) of each real-power generator and their new bus

TABLE I
GENERATOR DATA

Bus No.	1	2	5	8	11	13
P_G (MW)	94	72	48	35	18	17
New Bus No.	1	2	3	6	5	4

TABLE II
SIMULATION TIME

Layer	Step	Simulation Time (sec.) Each Step	Simulation Time (sec.) Each Layer
Off-line layer	Build $D((A^*_G)_{ij})$, $i=1,\ldots,N_G, j=1,\ldots,N$	0.750	
Periodic layer	Build $DV([[\lambda \cdot w_i^j]]), j=1,\cdots,N$	<0.001	≈ 15.6 (by 1 CPU) ≈ 7.0 (by 6 CPUs)
Periodic layer	Build $DV(SW_1)$	1.500	≈ 15.6 (by 1 CPU) ≈ 7.0 (by 6 CPUs)
Periodic layer	Build $DV(SW_2)$	1.531	≈ 15.6 (by 1 CPU) ≈ 7.0 (by 6 CPUs)
Periodic layer	Build $DV(SW_3)$	1.641	≈ 15.6 (by 1 CPU) ≈ 7.0 (by 6 CPUs)
Periodic layer	Build $DV(SW_4)$	2.203	≈ 15.6 (by 1 CPU) ≈ 7.0 (by 6 CPUs)
Periodic layer	Build $DV(SW_5)$	2.063	≈ 15.6 (by 1 CPU) ≈ 7.0 (by 6 CPUs)
Periodic layer	Build $DV(SW_6)$	1.922	≈ 15.6 (by 1 CPU) ≈ 7.0 (by 6 CPUs)
Periodic layer	Build $D(PBC)$	4.781	≈ 15.6 (by 1 CPU) ≈ 7.0 (by 6 CPUs)
On-line layer	Build $D(SSC)$	0.015	0.015
On-line layer	Build $D(BP)$	<0.001	0.015
On-line layer	Find all solutions	<0.001	0.015

numbers in G^r is shown in Table I. All other data are standard IEEE 30-bus data.

We select $d = 5\,\mathrm{MW}$, $\lambda = 10$, and safety coefficient $\alpha = 0.9$. From Fig. 6 and (26), we selected $n_b = 11$. Assume that $I_A = \{8, 11\}$ and $I_S = \{1, 2, 5, 13\}$ after a loss of synchronization is caused. We determine the variable ordering of OBDDs by the approach in Section III-C and perform simulations of three time layers. Simulation time in each computing step is shown in Table II.

Finally, there are a total of 252 solutions of BP problem (27 solutions are cut sets of G). Simulation results show that for a graph with $N < 30$ and $M < 30$, its BP problem can be quickly solved by our OBDD-based method.

B. Simulation Results of IEEE 118-Bus Network

G is shown in Fig. 10. We solve its SS problem. The generation (P_G) of each real-power generator is shown in Table III and the steady security limit (PSL) of each transmission line is shown in Table IV. Here, we assume that PSL's have only four capability classes: 65, 130, 300, and 600 MW. All other data are same as standard IEEE 118-bus data [31]. Then the SS problem of this IEEE 118-bus network is solved by phases. We first assume that, as shown in Table V, the loss of synchronization of generators in the same group do not happen because of their tight coupling. It follows that $N_G = 3$.

In fact, as discussed in Section III-E, the magnitude of N_G, how many groups we divided the generator nodes of G into, will not affect on the performance of our method because $DV(SW_1)$–$DV(SW_{N_G})$ can be concurrently constructed by N_G computers (or CPU's). Assuming $P_{\mathrm{Island}} = 500$ MW and $\sigma \approx 5\%$ after system splitting, it follows from (6) that

$$d_{\max} = \frac{\Delta f \cdot P_{\mathrm{Island}}}{\sigma \cdot f_0} = \frac{1 \times 500}{0.05 \times 60} = 167 \text{ (MW)}.$$
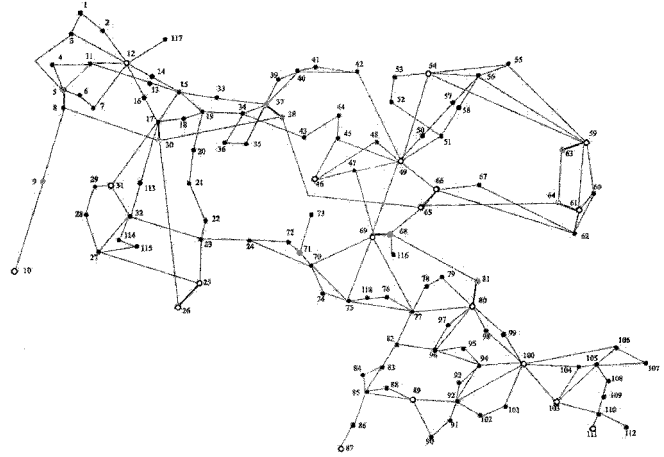

Fig. 10. Graph-model of IEEE 118-bus power network.

TABLE III
GENERATOR DATA

Bus No.	P_G (MW)	Bus No.	P_G (MW)	Bus No.	P_G (MW)
10	450.0	54	48.0	87	4.0
12	85.0	59	155.0	89	607.0
25	220.0	61	160.0	100	252.0
26	314.0	65	391.0	103	40.0
31	7.0	66	392.0	111	36.0
46	19.0	69	516.0		
49	104.0	80	477.0		

TABLE IV
TRANSMISSION LINE DATA

PSL(MW)	Transmission lines
600	8-9, 5-8, 9-10, 23-24, 25-26, 17-30, 8-30, 26-30, 37-38, 30-38, 38-65, 64-65, 65-68, 68-69
300	18-19, 19-20, 15-19, 20-21, 21-22, 22-23, 23-25, 25-27, 23-32, 15-33, 19-34, 33-37, 34-37, 60-61, 59-63, 63-64, 61-64, 49-66, 49-66, 69-70, 24-70, 70-71, 24-72, 71-72, 70-74, 70-75, 69-75, 74-75, 69-77, 75-77, 77-80, 77-80, 68-81, 80-81, 89-92, 89-92, 100-103, 68-116
130	4-5, 3-5, 5-6, 4-11, 5-11, 15-17, 17-18, 17-31, 27-32, 37-39, 37-40, 39-40, 40-41, 43-44, 34-43, 44-45, 47-49, 42-49, 42-49, 45-49, 48-49, 49-50, 49-51, 59-61, 47-69, 49-69, 76-77, 78-79, 79-80, 82-83, 83-85, 85-86, 86-87, 85-88, 85-89, 88-89, 89-90, 89-90, 92-93, 92-94, 80-96, 80-97, 80-98, 80-99, 92-100, 94-100, 100-104, 103-104, 103-105, 100-106, 104-105, 103-110, 110-112, 17-113, 32-113, 75-118, 76-118
65	1-2, 1-3, 6-7, 11-12, 2-12, 3-12, 7-12, 11-13, 12-14, 13-15, 14-15, 12-16, 16-17, 27-28, 28-29, 29-31, 31-32, 35-36, 35-37, 34-36, 40-42, 41-42, 45-46, 46-47, 46-48, 51-52, 52-53, 53-54, 49-54, 49-54, 54-55, 54-56, 55-56, 56-57, 50-57, 56-58, 51-58, 54-59, 56-59, 56-59, 55-59, 59-60, 60-62, 61-62, 62-66, 62-67, 65-66, 66-67, 71-73, 77-78, 77-82, 83-84, 84-85, 90-91, 91-92, 93-94, 94-95, 82-96, 94-96, 95-96, 96-97, 98-100, 99-100, 100-101, 92-102, 101-102, 105-106, 105-107, 105-108, 106-107, 108-109, 109-110, 110-111, 32-114, 27-115, 114-115, 12-117

TABLE V
GENERATOR GROUPS

Group No.	Node Serial Numbers	Total real-power generation (MW)
1	10, 12, 25, 26, 31	1076
2	46, 49, 54, 59, 61, 65, 66, 69, 80	2262
3	87, 89, 100, 103, 111	939

TABLE VI
MAPPING NODES

Index	Original Node Serial Numbers	Weight	New Node Serial Number
1	1~12, 16, 17, 25, 26, 30, 31, 117	612.0	1
2	46~69, 80, 81, 98, 116	887.4	2
3	85~92, 99~112	200.0	3
4	13~15	-38.0	5
5	18, 19	-105.0	6
6	32, 113~115	-95.0	15
7	27~29	-112.0	7
8	41, 42	-133.0	8
9	39, 40	-93.0	13
10	33~38	-146.0	4
11	43~45	-87.0	9
12	93~97	-137.0	10
13	82~84	-85.0	12
14	78, 79	-110.0	11
15	76, 77, 118	-162.0	16
16	74, 75	-115.0	17
17	70~73	-84.0	18
18	20~24	-62.0	14

TABLE VII
SIMULATION TIME

Layer	Step	Simulation Time (sec.)	
		Each Step	Each layer
Off-line layer	Build $D((A^*_G)_{ij})$, $i=1,...,N_G$, $j=1,...,N$	8.063	
Periodic layer	Build $DV([[\lambda \cdot w^r_i]]), i=1,\cdots,N$	<0.001	≈33.1 (by 1 CPU)
	Build $DV(SW_1)$	8.656	
	Build $DV(SW_2)$	10.750	
	Build $DV(SW_3)$	9.237	≈15.2 (by 3 CPUs)
	Build $D(PBC)$	4.469	
On-line layer	Build $D(SSC)$	0.094	
	Build $D(BP)$	0.016	≈0.14+0.007x (by 1 CPU)
	Find all solutions	0.031	
	Check RLC for 1 solution	0.007	
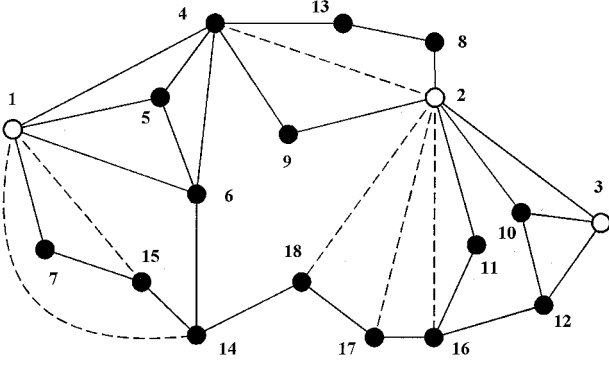
Fig. 11. Graph-model of reduced network.

Fig. 12. Two proper splitting strategies.

If we let $P_{\text{Area}} < d_{\max}$, then the whole power network can be divided into 18 areas according to P_{Area} and its characteristic as shown in Table VI. The nodes in same area are combined into a new node to construct G^r. (*SM-2* is performed). The mapping relation between new and old node serial numbers is also given in Table VI, where the new node serial numbers are set according to the approach in Section III-C.

G^r is shown in Fig. 11 ($N = 18$ and $M = 26$), where the broken lines denote the edges removed by *SM-1*. The size of searching space is decreased to 2^{26} after *Phase-1*. According to the approach in Section III-C, the ordering of corresponding 26 Boolean variables is set as

$$b_{1,4} < b_{1,5} < b_{1,6} < b_{1,7} < b_{2,3} < b_{2,8}$$
$$< b_{2,9} < b_{2,10} < b_{2,11} < b_{3,10} < b_{3,12} < b_{4,5}$$
$$< b_{4,6} < b_{4,9} < b_{4,13} < b_{5,6} < b_{6,14}$$
$$< b_{7,15} < b_{8,13} < b_{10,12} < b_{11,16}$$
$$< b_{12,16} < b_{14,15} < b_{14,18} < b_{16,17} < b_{17,18}.$$

Then, $D((A^*_G)_{ij})$ $(i = 1,\ldots,N_G,\ j = 1,\ldots,N)$ can be built based on this variable ordering. We select $d = 100$ MW, $\lambda = 10$ and safety coefficient $\alpha = 0.9$. From (26), we selected $n_b = 15$. Assuming that after a loss of synchronization

is caused, $I_A = \{1\}$ and $I_S = \{2,3\}$. We perform simulations of three time layers. Simulation time in each computing step is shown in Table V. In online layer, we check RLC for the former x solutions of BP problem by power-flow calculations. From Table VII, if we only use 1 CPU to perform the solving process, the total online computing time for finding a proper splitting strategy is generally less than 1 s. In this simulation process, there are total 48 120 solutions of BP problem (21 solutions are cut sets of G). After the former 1000 solutions are checked by power-flow calculations, 448 solutions (three solutions are cut sets of G) are given as proper splitting strategies at last. Two proper splitting strategies are shown in Fig. 12, where $\Delta(\Delta \le d)$ denotes the actual maximal real-power balance error

of two islands after system splitting. In the first splitting strategy, a cut set of G is cut off; in the second splitting strategy, in order to guarantee RLC, edges $e_{11,13}$, $e_{12,14}$, $e_{15,17}$, and $e_{17,18}$, which do not react on system splitting, are also cut off. Of course, a cut set splitting strategy, which generally leads to fewer transmission lines cut off, is top-priority for system splitting.

Comparing the simulation results on two networks, there is no distinct difference in time performance. That is because the sizes (N and M) of two reduced networks are similar. Combining the analyzes in Section III, we conclude that if we simplify a large-scale power network to a much simpler network with few enough nodes and edges (for example, by the computer resources applied in the above simulations, $N < 30$ and $M < 30$ are recommended), proper splitting strategies of this large-scale power system can be given in real-time by our OBDD-based method. Generally, it is not difficult to obtain that simpler network by *SM-1* and *SM-2*.

V. CONCLUSIONS

This paper uses an OBDD-based three-phase method to solving SS problems of large-scale power systems. Then a time-based layered structure of problem solving process is proposed to make this method more efficient for large-scale power systems, by which real-time decision-making is enabled and decision support systems for system splitting can be designed. Simulation results and further analyzes indicate that this method is effective for larger-scale power systems. Moreover, other OBDD-based algorithms and other decision graph representations can be attempted in SS problems.

ACKNOWLEDGMENT

REFERENCES

[1] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Trans. Comput.*, vol. C-35, pp. 677–691, Aug. 1986.
[2] ——, "Symbolic Boolean manipulation with ordered binary decision diagrams," *ACM Computing Surveys*, vol. 24, no. 3, pp. 293–318, Sept. 1992.
[3] Q. Zhao, K. Sun, and D. Zheng et al., "A Study of System Splitting Strategies for Island Operation of Power System: A Two-Phase Method Based on OBDDs," IEEE Trans. Power Syst., to be published.
[4] J. Thapar, V. Vittal, W. Kliemann, and A. A. Fouad, "Application of the normal form of vector fields to predict interarea separation in power systems," *IEEE Trans. Power Syst.*, vol. 12, pp. 844–850, May 1997.
[5] V. Vittal, W. Kliemann, Y.-X. Ni, D. G. Chapman, A. D. Silk, and D. J. Sobajic, "Determination of generator groupings for a islanding scheme in the manitoba hydro system using the method of normal forms," *IEEE Trans. Power Syst.*, vol. 13, pp. 1345–1351, Nov. 1998.
[6] J. L. Sancha, M. L. Llorens, J. M. Moreno, B. Meyer, J. F. Vernotte, W. W. Price, and J. J. Sanchez-Gasca, "Application of long-term simulation programs for analysis of system islanding," *IEEE Trans. Power Syst.*, vol. 12, pp. 189–197, Feb. 1997.
[7] H. B. Ross, N. Zhu, J. Giri, and B. Kindel, "An AGC implementation for system islanding and restoration conditions," *IEEE Trans. Power Syst.*, vol. 9, pp. 1399–1410, Aug. 1994.
[8] S. Agematsu, S. Imai, R. Tsukui, H. Watanabe, T. Nakamura, and T. Matsushima, "Islanding protection system with active and reactive power balancing control for Tokyo metropolitan power system and actual operational experiences," in *Proc. Inst. Elect. Eng. Seventh Int. Conf. Develop. Power Syst. Protection*, 2001, pp. 351–354.
[9] B. A. Archer and J. B. Davies, "System islanding considerations for improving power system restoration at manitoba hydro," in *Proc. IEEE Canadian Conf. Electr. Comput. Eng.* , vol. 1, 2002, pp. 60–65.
[10] M.-S. Tsai, "Development of islanding early warning mechanism for power systems," in *IEEE Power Eng. Soc. Summer Meeting*, vol. 1, 2000, pp. 22–26.
[11] K. Rajamani and U. K. Hambarde, "Islanding and load shedding schemes for captive power plants," *IEEE Trans. Power Delivery*, vol. 14, pp. 805–809, July 1999.
[12] O. I. Elgerd, *Electric Energy Systems Theory*, Second ed: McGraw-Hill., 1982.
[13] (1967) Prevention of Power Failures. U.S. Federal Power Commission, Government Printing Office, Washington, D.C., U.S. [Online]. Available: http://chnm.gmu.edu/blackout/archive/a_1965.html
[14] (1978) The Con Edison Power Failure U.S. Department of Energy. Federal Energy Regulatory Commission. Government Printing Office, Washington, DC, U.S. [Online]. Available: http://chnm.gmu.edu/blackout/archive/a_1977.html
[15] C. W. Taylor and D. C. Erickson, "Recording and analyzing the July 2 cascading outage [Western USA power system]," *IEEE Comput. Applicat. Power*, vol. 10, pp. 26–30, Jan. 1997.
[16] M. Thaden, "Analysis of a major load island outage on the POTOMAC electric power company system," *IEEE Trans. Power Syst.*, vol. 14, pp. 306–311, Feb. 1999.
[17] X. Vieira et al., The March 11:th 1999 Blackout: Short-Term Measure to Improve System Security and Overview of the Reports Prepared by the International Experts CIGRÉ Session, SC 39 Workshop on Large Disturbances, Paris, Aug. 29, 2000.
[18] M. Jonsson, "Line Protection and Power System Collapse," Chalmers Univ. Technology, School of Electrical and Computer Engineering, Göteborg, Sweden, 393L, 2001.
[19] C. H. Papadimitriou, *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
[20] K. H. Kim, *Boolean Matrix Theory and Applications*. New York: Marcel Dekker, 1982.
[21] V. Raman and A. N. Zamfirescu, "OBDD extraction from vhdl gate level descriptions at design elaboration," in *Proc. Fall VHDL Int. Users Forum Workshop*, 1999, pp. 30–39.
[22] F.-M. Yeh and S.-Y. Kuo, "OBDD-based network reliability calculation," *Electron. Lett.*, vol. 33, pp. 759–760, Apr. 1997.
[23] P. Mateu-Villarroya and J. Prades-Nebot, "Lossless image compression using ordered binary-decision diagrams," *Electron. Lett.*, vol. 37, pp. 162–163, Feb. 2001.
[24] J. R. Burch, E. M. Clarke, K. L. McMillan, D. L. Dill, and L. H. Hwang, "Symbolic model checking: 10^{20} states and beyond," in *Proc. 5th Annu. IEEE Symp. Logic Comput. Sci.*, 1990, pp. 428–439.
[25] Jørn Lind-Nielsen's BuDDy Package. [Online]. Available: http://www.itu.dk/research/buddy/index.html
[26] K. L. McMillan. The SMV System for SMV Version 2.5.4. [Online]. Available: http://www-2.cs.cmu.edu/~modelcheck/smv/smvmanual.ps
[27] Carnegie Mellon's BDD Package. [Online]. Available: http://www-2.cs.cmu.edu/afs/cs/project/modck/pub/www/bdd.html
[28] Fabio Somenzis's CUDD Package. [Online]. Available: http://vlsi.colorado.edu/~fabio
[29] Stefan Höreth's TUD DD Package. [Online]. Available: http://www.rs.e-technik.tu-darmstadt.de/~sth/download.html
[30] ABCD Package. [Online]. Available: http://iseran.ira.uka.de/~armin/abcd
[31] Ieee118cdf.txt [Online]. Available: http://www.ee.washington.edu/research/pstca/

Kai Sun was born in Heilongjiang, China, in 1976. He received the B.S. degree in automatic control from Tsinghua University, Beijing, China, in 1999. He is currently pursuing the Ph.D. degree in the Department of Automation at Tsinghua University.

His research interests include discrete event dynamic systems, hybrid systems, and power systems.

Da-Zhong Zheng received the diploma in automatic control from Tsinghua University, Beijing, China, in 1959.

Since 1959, he has been with the Department of Automatic Control at Tsinghua University, where he is a Professor in control theory and engineering. He is also a Vice-Chairman of control theory technical committee for Chinese Association of Automation (CAA), a Deputy Editor-In-Chief of *Acta Automatica Sinica*, Beijing, China, and an editor of *Asian Journal of Control* (AJC), Taipei, Taiwan, R.O.C. He was a Visiting Scholar in Department of Electrical Engineering, State University of New York at Stony Brook, from 1981 to 1983 and in 1993. His research interests include linear systems, discrete event dynamic systems, and power systems.

Mr. Zheng has published many journal papers and five books. .

Qiang Lu (SM'85-F'03) graduated form the Graduate School of Tsinghua University, Beijing, China, in 1963.

Currently, he is a Professor in Tsinghua University, and an academician of Chinese Academy of Science since 1991. He was a visiting Scholar and a visiting professor in Washington University, St. Louis, MO, and Colorado State University, Ft. Collins, from 1984 to 1986. He was also a Visiting Professor at Kyushu Institute of Technology (KIT), Japan, from 1993 to 1995. His research interest is in nonlinear control theory application in power system and digital power systems.