Searching for feasible splitting strategies of controlled system islanding

K. Sun, D.-Z. Zheng and Q. Lu

Abstract: Controlled system islanding, also called system splitting, can effectively prevent blackouts of power systems. The splitting strategy determining how to split a power network into islands should be given in real time. Previous papers proposed an ordered binary decision diagram method to search for the splitting strategies satisfying necessary steady-state constraints, e.g. generation–load balance and transmission-line capacity constraint, and also showed that, without any other corrective controls, the splitting strategies may produce unstable islands according to a further simulation study. A modified method to find feasible splitting strategies in real time for large power network is presented. Each feasible splitting strategy not only satisfies necessary steady-state constraints but also easily produces stable islands to prevent a blackout. The method also introduces new techniques into strategy searching to increase its efficiency and practicality, e.g. network partitioning and parallel processing, generator classifying, deriving splitting strategies from cut-set splitting strategies, etc. Simulations on the IEEE 118-bus system show that the real-time portion of the modified method finds feasible splitting strategies in less than one second.

Abbreviations

CG	Crucial generator
CSC	Cut-set constraint
NCG	Noncrucial generator
OBDD	Ordered binary decision diagram
PBC	Generation-load balance constraint
RLC	Transmission-line capacity constraint
SSC	Constraint that asynchronous groups of generators must be separated
TVC	Threshold value constraint

1 Introduction

In a power network serious faults may degrade its stability and cause oscillation and even loss of synchronism between groups of generators. If generators cannot be efficiently resynchronised and stabilised, passive islanding can occur following a series of relay actions. Unfortunately, passive islanding often produces generation–load unbalanced or unstable electrical islands, which perhaps continue to collapse until blackout. Some papers have studied controlled system islanding [1–7], or system splitting, which means that the dispatch centre actively trips some lines to split the power network into several maintainable islands according to asynchronous groups of generators and other requirements. It can prevent a blackout and maintain electricity supply for most customers, although the power network will be separated into asynchronous islands.

However, it is not easy in real-time to determine the splitting strategy, namely which lines should be tripped, when system splitting is imperative and asynchronous groups of generators have been detected. References [1–3] have made some effort to solve this problem. Its main difficulties lie in the following aspects. First, real-time decision-making requires extremely short strategy-search time (generally, hundreds of milliseconds), but the strategy space will explode exponentially with the increasing of size and complexity of the power network [2]. Secondly, the splitting strategy should satisfy necessary steady-state constraints, e.g. the following three constraints proposed in [2]:

- asynchronous groups of generators must be separated (denoted by SSC),
- generation–load imbalance in each island must be less than a prescribed limit (denoted by PBC),
- all lines in each island must be loaded below their steady-state transmission capacity limits (denoted by RLC).

Thirdly, the splitting strategy should ensure the stability of every island produced by system splitting.

Reference [2] proposed a graph-model to represent a power network by which graph theory and boolean algebra can be applied to represent and analyse splitting strategies. Based on ordered binary decision diagram (OBDD) representation [8], which is a high-efficiency technique for solving complicated boolean algebra problems, [1] proposed a three-phase method to find the splitting strategies satisfying SSC, PBC and RLC in real-time. A time-based layered structure was also proposed to realise the method's real-time searching. However, not all splitting strategies given by the method can ensure the stability of every island, as shown in the simulation study of [3]. That is because the method only focuses on steady-state constraints to enable

IEE Proceedings online no. 20050168
doi:10.1049/ip-gtd:20050168
Paper first received 6th May 2005 and in final revised form 28th August 2005

K. Sun and D.-Z. Zheng are with the Department of Automation, Tsinghua University, Beijing, 100084, China
Q. Lu is with the Department of Electrical Engineering, Tsinghua University, Beijing, 100084, China
E-mail: sunkai76@gmail.com

fast decision-making, and it assumes that available corrective control measures can stabilise each island after system splitting. To improve the method, paper [3] introduces a new constraint, 'threshold value constraint' (TVC), to exclude the splitting strategies that may cause a large power-flow fluctuation and probably produce unstable islands. Simulations show that if the threshold values of TVC are properly selected, almost all splitting strategies satisfying SSC, PBC, RLC and TVC can split the power network into stable and maintainable islands. Currently the method is still in a theoretical and trial stage. Before applcation to actual power systems it requires further study, e.g. improving its time performance for large power networks, increasing the feasibility of splitting strategies (i.e. preventing or decreasing unstable islands), considering more practical factors of power systems, making better use of the inherent structural characteristics of a power network, introducing parallel techniques, etc.

This paper presents a modified method considering these aspects. First, the modified method takes account of grid loss to construct a zero-weight-sum graph-model of the power network. Secondly, an offline network partitioning technique is proposed to make parallel processors simultaneously start real-time strategy searching in all subnetworks. Thirdly, generators are classified and differently treated according to importance and capacity. Fourthly, a deriving relation between splitting strategies is defined, which suggests the order of strategy searching and checking. Finally, TVC is introduced into the modified method to exclude the splitting strategies that produce unstable islands. Thus the splitting strategies satisfying SSC, PBC, RLC and TVC are first considered in system splitting, and are reasonably called 'feasible splitting strategies' in this paper. Analysis and simulation results on the IEEE 118-bus system show that, for arbitrarily assumed asynchronous groups of generators, the modified method can give feasible splitting strategies in real time.

2 Comparison between two methods

Each phase's tasks of the original method and the modified method are listed and compared in Table 1, where the words in bold indicate their differences. A cut-set splitting strategy is a cut set of the power network's graph-model [2], which only includes the lines contributing to separate islands. More details are given in the following Section. Offspring strategies of a splitting strategy is also defined.

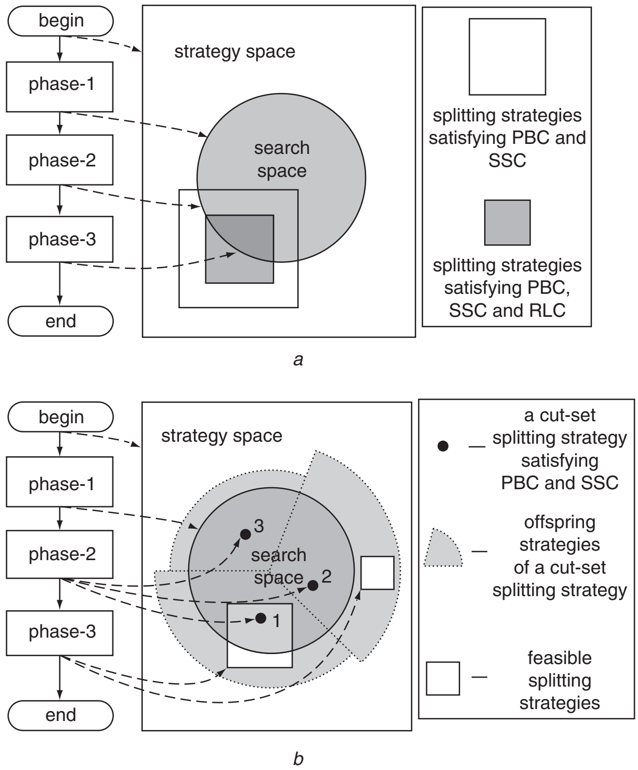


Fig. 1 *Comparison between search processes of two methods*
a Original method
b Modified method

The idea of the original method is illustrated in Fig. 1*a*. The larger white rectangle denotes the original strategy space, whose size equals 2^M if the power network has M lines. The grey circle denotes the search space created by phase-1, which is the actual searching scope of the original method and whose size can be set much smaller by preprocessing measures. The larger white square denotes the set of all strategies satisfying PBC and SSC in the strategy space, and the smaller grey one within it denotes the strategies also satisfying RLC. The splitting strategies found by the original method are denoted by the intersection of the small grey square and the grey circle.

Figure 1*b* illustrates the modified method's searching process by a typical case. The larger white rectangle and the grey circle have the same meanings as those in Fig. 1*a*.

Table 1: Comparison between original and modified methods

Phases	Original method	Modified method
	Initialise parameters and decide search space by preprocessing measures:	Initialise parameters and decide search space by preprocessing measures:
1	• Construct graph-model	• Construct **zero-weight-sum** graph-model **considering grid loss**
	• Combine nodes by areas and reduce irrelevant nodes and edges	• **Partition power network into subnetworks**
		• **Classify generators**
		• Combine nodes by areas and reduce irrelevant nodes and edges **in each subnetwork**
2	Find all splitting strategies satisfying PBC and SSC in search space by OBDDs	Find all **cut-set splitting** strategies satisfying PBC and SSC in search space by OBDDs and **parallel processing**
3	Check RLC for splitting strategies until strategy satisfying SSC, PBC and RLC found	**Orderly check TVC and RLC for cut-set splitting strategies and their offspring strategies until feasible splitting strategy is found**

Suppose that there are totally three cut-set splitting strategies satisfying SSC and PBC in the search space, as shown by the three black dots numbered 1–3. They can be found by phase-2. The three sectors covering three black dots denote all offspring strategies generated by three cut-set splitting strategies. Assume that only strategies 1 and 2 have offspring strategies satisfying SSC, PBC, RLC and TVC, and strategy 1 itself satisfies all the four constraints. That situation is shown by two small squares in Fig. 1*b*, which are just the feasible splitting strategies found in phase-3. Compared with the original method, the modified method's searching scope is not limited in the search space but covers all offspring strategies of the cut-set splitting strategies found in phase-2. In Fig. 1*b* the actual searching scope is the combination of the three grey sectors.

3 New characteristics of modified method

3.1 Zero-weight-sum graph-model

As discussed in [3], it is reasonable to focus only on the backbone grid (e.g. 220 or 500 kV rating) of a large power network when we study its splitting strategies. Meanwhile, we use some equivalent generators to replace main power plants. Thus in the rest of the paper, the power network generally means the backbone grid of a large power network.

A graph-model $G(V,E,W)$ [2] depicts a n-bus power network, where node set $V=\{v_1, \ldots, v_n\}$ and edge set E with elements $e_{ij}(i<j)$, respectively, denote buses and transmission lines, and node weight set $W=\{w_1, \ldots, w_n\}$ is determined by injected real powers from buses. A splitting strategy can be depicted by an edge set $S \subset E$. As an example, Fig. 2 shows the IEEE 118-bus system's graph-model, where white dots are generator nodes denoting the buses where generators are installed, and black dots are load nodes denoting the other buses.

In the original method, node weights are defined as injected real powers of corresponding buses. Since grid loss exists, (1) does not hold, i.e. the graph-model is not zero-weight-sum. The modified method constructs a zero-weight-sum graph-model G satisfying (1) in phase-1 to make better use of the conclusions about zero-weight-sum graphs. For instance, for the most common case of two-island splitting, checking only the PBC of any one island is enough if G is zero-weight-sum

$$\sum_{i=1}^{n} w_i = 0 \tag{1}$$

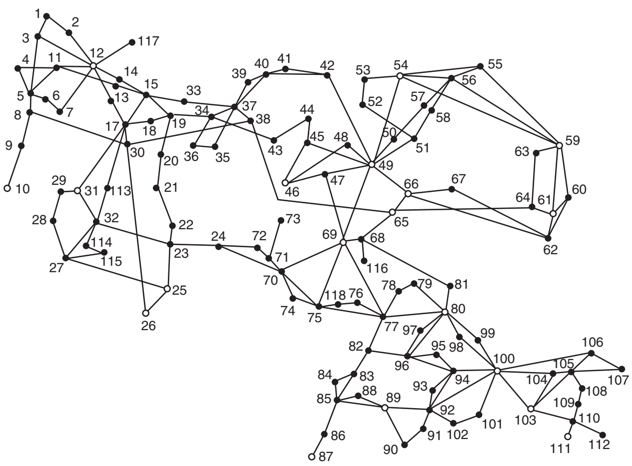


Fig. 2 *IEEE 118-bus system's graph-model*

First, estimate total grid loss P_{loss} by (2), where P_i, $P_{G,i}$ and $P_{L,i}$ are, respectively, injected real power, real generation power, and real load power of bus i

$$P_{loss} = \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} (P_{G,i} - P_{L,i}) \tag{2}$$

Then node v_is weight w_i can be determined by either (3) or (4) to satisfy (1). Equation (3) subtracts grid loss proportionately from all real power outputs; (4) adds grid loss proportionately to all real power inputs

$$w_i = \begin{cases} P_i\left(1 - P_{loss}\Big/\left|\sum_{P_i>0} P_i\right|\right), & \text{if } P_i > 0 \\ P_i, & \text{if } P_i \le 0 \end{cases} \tag{3}$$

$$w_i = \begin{cases} P_i, & \text{if } P_i \ge 0 \\ P_i\left(1 + P_{loss}\Big/\left|\sum_{P_i<0} P_i\right|\right) & \text{if } P_i < 0 \end{cases} \tag{4}$$

In fact, PBC, RLC and TVC do not demand too accurate values of node weights, so in the modified method node weights are calculated offline from typical generation and load data and are updated online only if power flow obviously fluctuates.

We do not consider reactive power balance in this paper since real power balance is more crucial to system splitting. Moreover, reactive power imbalance can be compensated by local reactive power compensators in practice. The reactive power balance problem in system splitting is not the emphasis of this paper, but will be considered in our future research.

3.2 Deriving relation between splitting strategies

We now study a kind of intrinsic deriving relation between splitting strategies. Using it can obviously increase the efficiency of the strategy search. Suppose that two splitting strategies S_1 and S_2 ($S_1, S_2 \subset E$) produce the same number of islands, and there is $S_1 \subseteq S_2$. We say S_2 is an offspring strategy of S_1, and S_1 is a parent strategy of S_2. Obviously, S_1 is an offspring and parent strategy of itself. Moreover, if S_2 has k more elements (i.e. edges) than S_1, we say S_2 is a k-level offspring strategy of S_1 and, naturally, S_1 is a k-level parent strategy of S_2. Since a splitting strategy usually has many offspring strategies at the same level, we use $(S)_i^k$ to denote the ith k-level offspring strategy of splitting strategy S. Obviously, every cut-set splitting strategy has only offspring strategies but no parent strategy except itself. This analysis shows that every splitting strategy can be derived from a cut-set splitting strategy. Therefore the modified method's phase-2 is focused on searching for cut-set splitting strategies satisfying SSC and PBC.

To obtain cut-set strategies we introduce cut-set constraint (CSC): if a line is cut-off after system splitting, its two terminal buses must belong to different islands. CSC can be offline expressed by OBDDs in phase-2.

3.3 Network partitioning

Power system network partitioning [9] is an important technique associated with the application of parallel processing. Its objective is to divide the whole power network into subnetworks, which are simultaneously managed by parallel processors. That can enhance the efficiency of power system analysis, planning and operation.

Introducing a similar idea into the searching of splitting strategies, the modified method offline partitions a large power network into N subnetworks in phase-1. Consequently constraints SSC, CSC and PBC are rebuilt in each subnetwork. In phase-2, the searching process for the splitting strategies satisfying the three constraints is partitioned into N parallel procedures. That will greatly increase the searching speed. Then N subnetworks' cut-set splitting strategies are incorporated to form the power network's cut-set splitting strategies. Finally, TVC and RLC are checked. This Section mainly studies how to partition a power network. The other problems are studied in Sections 3.4–3.6.

Generally a power network can be naturally partitioned according to its structural characteristics or actual geographic areas. Here we propose a network-partitioning approach based on graph theory to partition the power network's graph-model G into a number of subgraphs denoting as many subnetworks. To avoid arduous analysis work in incorporating all subnetworks' splitting strategies, we demand that all adjacent ones of the subgraphs jointly cover an interface graph having as few load nodes as possible. The approach has four steps

(i) In $G(V,E,W)$, select an interface graph denoted by $G_0(V_0,E_0,W_0)$ which partitions the rest of G into K subgraphs $G_1(V_1,E_1,W_1)$–$G_K(V_K,E_K,W_K)$ as shown in Fig. 3*a*. G_0 should have as few load nodes as possible. The following three conditions are satisfied: first, $V_1 \cup V_2 \cup \cdots \cup V_K \cup V_0 = V$ and $V_1 \cap V_2 \cap \cdots \cap V_K \cap V_0 = \phi$; secondly, if nodes v_i and v_j are both in one of G_0–G_K, then edge e_{ij} connecting v_i and v_j, is also in it; thirdly, there is no edge directly connecting any two of G_1–G_K.

(ii) Construct G's subgraphs $G_{SNk}(V_{SNk},E_{SNk},W_{SNk})$ ($k = 1$–K) satisfying $V_{SNk} = V_k \cup V_0$, and $E_{SNk} = E_k \cup E_0 \cup$ {all edges directly connecting G_{SNk} with G_0}. Thus G_{SN1}–$G_{SN,K}$ all contain the interface graph G_0 as shown in Fig. 3*a*.

(iii) Partition G_0's node weights into G_{SN1}–$G_{SN,K}$ according to the following rule: if a node v_i of G_0 has weight w_i, then its weights in $G_{SN,1}$–$G_{SN,K}$ (denoted by w_{i1}–w_{iK}) satisfy (5) and (6). Equation (6) makes each subgraph still zero-weight-sum.

$$\sum_{k=1,\ldots,K} w_{ik} = w_i, \quad \forall v_i \in G_0 \tag{5}$$

$$\sum_{v_l \in V_k} w_l + \sum_{v_i \in V_0} w_{ik} = 0, \quad k = 1 - K \tag{6}$$

(iv) Recursively apply steps (*a*)–(*c*) to partition still complex subgraphs. We assume that there are finally N subgraphs $G_{SNk}(V_{SNk},E_{SNk},W_{SNk})$ ($k = 1 - N$) and N_0 interface graphs $G_{0,1}$–$G_{0,N0}$.

As an example, Table 2 gives a possible way to partition the IEEE 118-bus system's graph-model into three subgraphs $G_{SN,1}$–$G_{SN,3}$ by two interface graphs $G_{0,1}$ and $G_{0,2}$, as shown in Fig. 3*b*, where $G_{SN,1}$ contains G_1, $G_{0,1}$ and $G_{0,2}$, $G_{SN,2}$ contains G_2 and $G_{0,1}$, and $G_{SN,3}$ contains G_3 and $G_{0,2}$. An advantage of this partition is that every interface graph only has generator nodes. Node weights $w_{100,3}$, $w_{49,2}$ and $w_{65,2}$ are first determined to make $G_{SN,2}$ and $G_{SN,3}$ a zero-weight-sum. Then $w_{100,1}$, $w_{49,1}$ and $w_{65,1}$ are calculated by

$$w_{100,1} = w_{100} - w_{100,3}; \quad w_{49,1} = w_{49} - w_{49,2}; \quad w_{65,1} = w_{65} - w_{65,2}$$

After the network partitioning, the largest subgraph G_1 has 91 nodes, which may either continue to be partitioned or be further reduced by the preprocessing measures in [1], e.g. combining nodes by areas or reducing irrelevant nodes and edges.

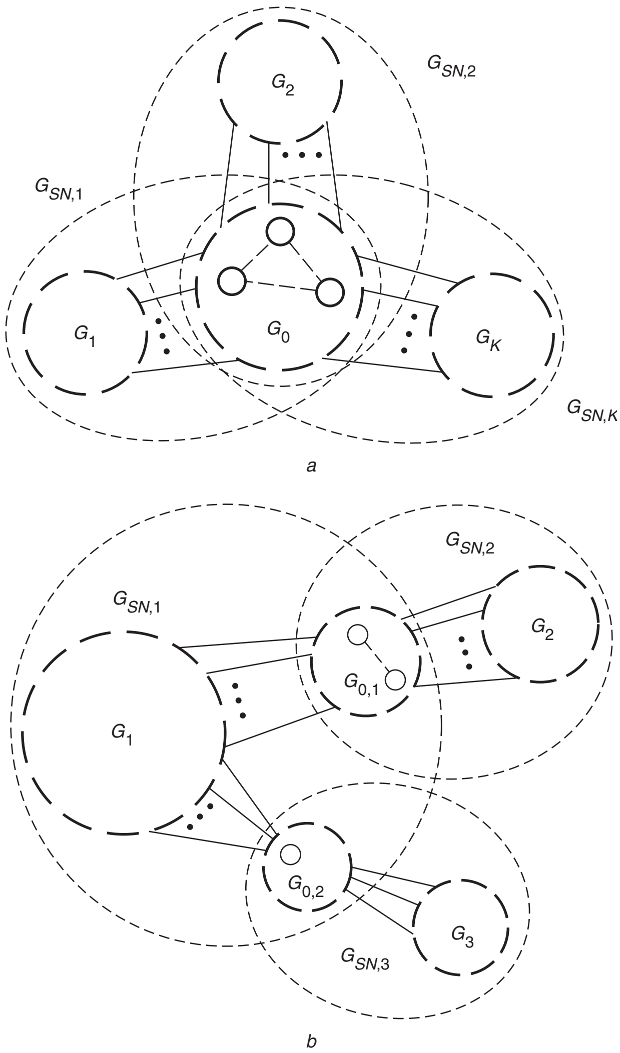


Fig. 3 *Approach to network partitioning*
a Key idea
b Feasible network partitioning of IEEE 118-bus system

Table 2: Partitioning the IEEE 118-bus system

Subgraphs	G_1	G_2	G_3	$G_{0,1}$	$G_{0,2}$
Node serial numbers	1–48, 68–99, 101, 102, 113–118	50–64, 66, 67	103–112	49, 65	100

3.4 Generator classifying

Some generators play more important roles than others in power system operation and control, so it is not advisable to treat all generators equally in splitting strategy searching. The modified method offline classifies generators of each subnetwork into two types: crucial generators (CGs): ones that have large generation capacities or important functions in operation and control, and non-crucial generators (NCGs) which are the others. Separating a few NCGs from the system generally does not affect its generation–load balance and stability. For example, in the simulations of [3], tripping several small-capacity generators does not affect the success of system splitting. Accordingly NCGs are treated as follows: if a NCG has adjacent load nodes it may be directly islanded with approximately matched load to form an island when necessary; otherwise it may be tripped. Hence only CGs are considered by OBDD-based algorithms [1] to search for splitting strategies. If a generator lies in an interface graph it may be regarded as a NCG of a subnetwork while it is classified as a CG of another subnetwork. That depends on its weights in respective subnetworks.

The preprocessing measure 'combining nodes by areas' in [1] needs to be modified to reflect this kind of classifying. Each subnetwork's nodes are offline divided into generator areas (NCG areas and CG areas) and load areas in phase-1. A NCG area comprises a NCG node and approximately matched load nodes if the NCG node has adjacent load nodes; otherwise, the NCG area is the NCG node itself. Since NCGs have a relatively weak influence on system stability and generation–load balance, OBDD-based algorithms do not consider NCG areas in the searching of each subnetwork's splitting strategies. Then we judge which island each NCG area belongs to when incorporating all subnetworks' splitting strategies. The other nodes are divided into CG areas and load areas. Each CG area contains a CG node, and may also contain some load nodes if the CG node has adjacent load nodes. However, we do not demand that these load nodes be matched since a CG has usually a large generation capacity. Finally, each load area has only load nodes. In the kth subnetwork, let $P_{Area,k}$ denote the absolute upper limit of a load area's weight sum, which is recommended to satisfy (7) [1]

$$P_{Area,k} \le \frac{\Delta f P_{Island}}{\sigma f_0} \triangleq d_{max} \qquad (7)$$

where d_{max} is the maximum acceptable generation–load imbalance in each island, P_{Island} is an estimated lower limit of each island's total real generation power, Δf is a prescriptive frequency-offset limit of each island, constant $\sigma = 2$–5%, and f_0 is the rating frequency. In general, d_{max} is relatively large, but it is still possible that a single load node's weight slightly exceeds d_{max}. Thus the load node itself should be regarded as a load area. In fact, this situation seldom exists especially in a large power network. In the power network's PBC we need to prescribe an upper limit (denoted by d) for each island's weight sum, which should satisfy $d \le d_{max}$.

Here we make each generator area contain only one generator node (perhaps corresponding to a power plant) to enable the modified method to deal with any possible asynchronous groups of generators. In practice, if some generators always keep coherent after faults occur, they may be put into the same generator area.

Then merge each of NCG areas, CG areas, and load areas into an equal NCG, CG or load node. Thus a reduced graph is formed from each subnetwork, where all NCG nodes are ignored. The graph is further reduced by the preprocessing measure 'reducing irrelevant nodes and edges' in [1]. Use $G^r_{SN,1} - G^r_{SN,N}$ to denote the final N reduced graphs which, respectively, correspond to the N subnetworks. For computer conditions similar to that used in the following simulations, we recommend that nodes and edges of each reduced graph should both be less than 40 to make OBDD-based algorithms more efficient.

3.5 Splitting strategy searching by parallel processing

As discussed in Section 3.2, the modified method's phase-2 searches for cut-set splitting strategies satisfying SSC and PBC in the searching space by OBDD-based algorithms. The search process has two steps: first, find all cut-set splitting strategies satisfying SSC and PBC in each of $G^r_{SN,1} - G^r_{SN,N}$, then incorporate them into the power network's splitting strategies satisfying SSC and PBC.

For each $G^r_{SN,k}$ let I_k be its node serial number set and $I_{G,k}$ be its CG-node serial number set. Assume that all CGs in the kth subnetwork are detected to separate into $N_{G,k}$ asynchronous groups. Use $I_{G,k,1} - I_{G,k,N_{G,k}}$ to denote the sets of their corresponding node serial numbers in $G^r_{SN,k}$. Obviously

$$I_{G,k} = I_{G,k,1} \cup I_{G,k,2} \cup \cdots \cup I_{G,k,N_{G,k}} \qquad (8)$$

Then $G^r_{SN,k}$'s CSC, PBC and SSC (denoted by CSC_k, PBC_k and SSC_k, respectively) are expressed in boolean functions as follows.

First, from the meaning of CSC, we have

$$CSC_k = \prod^{\otimes}_{\forall i,j \in I_K, i<j} \langle (A^*_{G,k})_{ij} \rightarrow b_{k,i,j} \rangle = \prod^{\otimes}_{\forall i,j \in I_K, i<j} \left[\overline{(A^*_{G,k})_{ij}} \oplus b_{k,i,j} \right] \qquad (9)$$

where $\rightarrow$, $\oplus$ and $\otimes$ respectively, denote logic operations 'implication', 'or', and 'and'; $A_{G,k}$ is $G^r_{SN,k}$'s adjacency matrix. boolean variable $b_{k,i,j} = (A_{G,k})_{ij} = (A_{G,k})_{ji}$ (assume $i<j$) equals 1 if nodes i and j are connected by an edge in $G^r_{SN,k}$; otherwise, it equals 0; $A^*_{G,k}$ is defined by

$$A^*_{G,k} \overset{\text{def}}{=} I \oplus A_{G,k} \oplus A^2_{G,k} \oplus \cdots \oplus A^L_{G,k} \qquad (10)$$

where I is the identity matrix, and L is the length of the longest path in $G^r_{SN,k}$ (for details, see [1]).

Secondly, referring to [1], we have

$$PBC_k = \prod^{\otimes}_{i \in I_{G,k}} \langle |(A^*_{G,k})_{i^*} \cdot W_k| \le d_k \rangle \qquad (11)$$

$$SSC_k = \prod^{\otimes}_{j=1,\ldots,N_{G,k}} \left[\prod^{\otimes}_{\forall i \in I_{G,k,j}} (A^*_{G,k})_{i,i_{G,k,j}} \right] \otimes \prod^{\otimes}_{\forall l \notin I_{G,k}} \left[(A^*_{G,k})_{l,i_{G,k,1}} \overline{\oplus} (A^*_{G,k})_{l,i_{G,k,2}} \overline{\oplus} \cdots \overline{\oplus} (A^*_{G,k})_{l,i_{G,k,N_{G,k}}} \right] \qquad (12)$$

where W_k is a column vector comprising $G^r_{SN,k}$'s all node weights, $\overline{\oplus}$ is logic operation exclusive-or, $i_{G,k,i} \in I_{G,k,i}$ is an arbitrarily selected serial number, and d_k is a prescribed absolute upper limit for each island's weight sum in the kth subnetwork. Considering the fact that an island may own several parts lying in different subnetworks and their weight sums may counteract when they are incorporated, we allow d_k to exceed d. After all subnetworks are incorporated, we use d to recheck each island's weight sum. It is recommended that $d \le d_k \le d_{max}$.

Then, after the priority ordering of all boolean variables $b_{k,i,j}$ is determined by the approach in [1], the OBDDs of CSC_k, PBC_k and SSC_k, denoted by $D(CSC_k)$, $D(PBC_k)$ and $D(SSC_k)$, can be built and then compose one OBDD, $D(CSC_k \otimes PBC_k \otimes SSC_k)$. Consequently $G^r_{SN,k}$'s all cut-set splitting strategies satisfying SSC and PBC are quickly found by OBDD-based algorithms.

The cut-set splitting strategies of subnetwork k (described by $G_{SN,k}$) can be determined from $G^r_{SN,k}$'s cut-set splitting strategies by means of the relations between edges of $G_{SN,k}$ and $G^r_{SN,k}$. Finally, the following steps will determine the power network's cut-set splitting strategies satisfying SSC and PBC.

(a) For each NCG area, if there is an adjacent island whose generators are synchronous with its NCG (i.e. they are in one coherent generator group), we reconnect the lines between them. Otherwise, isolate the NCG area as an individual island or trip its NCG if it has no load.
(b) Reconnect all synchronous and adjacent islands in the power network.
(c) If an interface graph's load nodes are contained by different islands, then we put each load node into the island where its weight is largest in absolute value. For example, if load node v_i's weights in $G_{SN,1}$–$G_{SN,N}$ are, respectively, w_{i1}–w_{iN}, and $|w_{i,l}| \geq |w_{i,k}|(\forall k = 1\text{–N})$, v_i will be put into $G_{SN,k}$.
(d) For each splitting strategy check every island's weight sum. If it exceeds d the splitting strategy will be excluded. This situation is caused by an increased accumulation of all subnetworks' generation–load imbalances.

The islands produced by a cut-set splitting strategy are perhaps greater than the detected asynchronous groups if the following two situations occur. First, generators of a coherent generator group lie in different subnetworks and are finally put into different nonadjacent islands, which are unable to combine to one island. Secondly, a NCG area is isolated as an individual island. However, these do not affect the feasibility of splitting strategies. Furthermore, if we compare the splitting strategies obtained by parallel processing and followed incorporation of subnetworks with the splitting strategies obtained by a direct whole network search, besides the fact that more islands may be produced as mentioned, the former may be fewer in number if d_k is so small as to exclude some strategies satisfying PBC.

3.6 Orderly checking TVC and RLC

TVC uses inequalities (13) and (14) to limit the disturbance degree caused by a splitting strategy S. Here inequality (14) has been adapted to make TVC easily checked by computer

$$\gamma_{Net}(S) \stackrel{\text{def}}{=} \frac{\sum\limits_{e_{ij}\in S} |P_{ij}|}{\sum\limits_{P_k>0,\, v_k\in V} P_k} \leq \Gamma_{Net} \quad (13)$$

$$\gamma_{Island}(S) \stackrel{\text{def}}{=} \max_{l=1,\ldots,N_I} \left(\frac{\sum\limits_{e_{ij}\in S, [A^*(S)]_{i,i_l} = [A^*(S)]_{j,i_l}=1} |P_{ij}|}{\sum\limits_{P_k>0,\, [A^*(S)]_{k,i_l}=1} P_k} \right) \leq \Gamma_{Island} \quad (14)$$

P_{ij} is the real transmission power of line i–j; N_I is the number of the islands produced by S; i_1–iN_I are N_I node serial numbers selected from N_I islands; $A(S)$ is the adjacency matrix of G after it is split by S; $[A(S)]_{ij} = 1$ if $\exists e_{ij} \in E\text{-}S$, or 0, otherwise; $A^*(S)$ can be calculated by the same way as (10); Γ_{Net} and Γ_{Island} are two selected threshold values. Paper [3] gives an approach to selecting proper Γ_{Net} and Γ_{Island}.

For two splitting strategies S_1 and S_2, assume that S_2 is the ith k-level offspring strategy of S_1, namely $S_2 = (S_1)_i^k$. Obviously, compared with S_1, S_2 trips k more lines but does not produce one more island, so we easily obtain

$$A^*(S_2) = A^*(S_1) \quad (15)$$

$$\gamma_{Island}(S_2) \geq \gamma_{Island}(S_1) \quad (16)$$

Two immediate conclusions are: a cut-set splitting strategy and its all offspring strategies have equal $A^*(\bullet)$ matrixes; for certain Γ_{Net} and Γ_{Island}, if a splitting strategy does not satisfy TVC then its all offspring strategies does not satisfy TVC either. Thus a natural idea is to first check TVC for each cut-set splitting strategy.

Moreover, checking TVC is much simpler than checking RLC since the latter needs power-flow calculation. Therefore the modified method checks RLC for a splitting strategy only if TVC has been satisfied. Once RLC is also satisfied the splitting strategy is given as a feasible splitting strategy.

Accordingly, the modified method's phase-3 is designed as follows. Suppose that phase-2 finds N_C cut-set splitting strategies satisfying SSC and PBC (denoted by $S_{C,1}$–$S_{C,Nc}$). A feasible splitting strategy can be given by the following search procedure:

- For each $S_{C,i}$ ($i = 1$–N_C), calculate $A^*(S_{C,i})$ and check TVC. Only if TVC is satisfied, RLC is checked. If TVC and RLC are both satisfied, output $S_{C,i}$ as a feasible splitting strategy and stop the procedure.
- If none of $S_{C,1}$–$S_{C,Nc}$ is feasible splitting strategy, continue the same checking in the last step for their offspring strategies according to the ordering 'from low level to high level', but ignore the splitting strategies whose parent strategies do not satisfy TVC. Stop the procedure until a feasible splitting strategy is found.

This search procedure is illustrated by the tree-like structure in Fig. 4. In phase-3, N_C parallel processors are used to simultaneously start N_C such searching procedures from N_C cut-set splitting strategies.

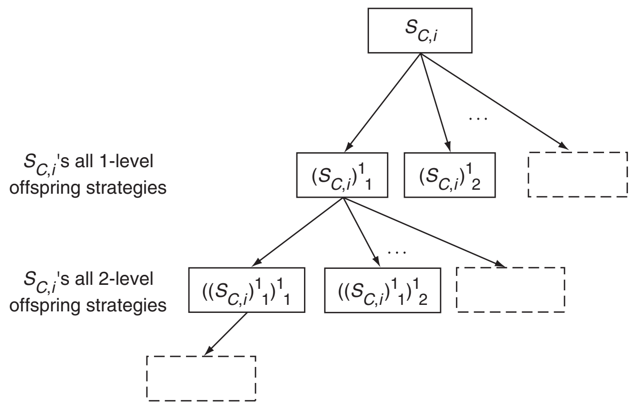


Fig. 4 *Tree structure for searching for feasible splitting strategy (i = 1 ~ NC)*

4 Simulation

4.1 Simulation object and data

The performance of the modified method is checked on the IEEE 118-bus system by a Pentium IV 2 GHz PC. Simulation models and data are the same as [3]. Simulations are performed based on a three-layer structure like that in [1] and all tasks of the modified method are divided into three time layers:

Offline layer: tasks are independent of online information and are performed offline. They includes all tasks of phase-1 and partial tasks of phase-2, e.g. expressing CSC_k by OBDDs, etc.

Online layer: tasks depend on online generation and load data and are performed with an interval of tens of minutes

Table 3: Node weights of IEEE 118-bus system

SN	10	12	25	26	49	61	65	66	69	80	87	89	100	103	111
w_i (MW)	433.9	36.6	212.1	302.8	112.9	154.3	377.0	340.4	497.9	334.6	3.9	585.3	207.3	16.4	34.7
P_i (MW)	450.0	38.0	220.0	314.0	117.0	160.0	391.0	353.0	516.4	347.0	4.0	607.0	215.0	17.0	36.0

to an hour. They include most tasks of phase-2, e.g. expressing PBC_k by OBDDs, etc.

Real-time layer: contains the other tasks, which need to be done in real-time after faults occur.

Simulations are performed according to the three time layers in Sections 4.2–4.4.

4.2 Offline layer

Set $\Gamma_{Net}=0.3$ and $\Gamma_{Island}=0.1$ by the approach in [3]. Then preprocess the power network. First, calculate total grid loss $P_{loss}=135.4$ MW from (2). We use (3) to calculate node weights. Table 3 gives the node weights w_i different from corresponding P_i. Partition the power network as shown in Table 2 and Fig. 3*b*, and also partition the node weights of $G_{0,1}$ and $G_{0,2}$ as $w_{49,1}=-8.4$, $w_{49,2}=121.3$, $w_{65,1}=377$, $w_{65,2}=0$, $w_{100,1}=-20.5$, $w_{100,3}=227.8$. Since most generators have a capacity larger than 150 MW, let $P_{Island}=150$ MW. Assume that $\sigma=2\%$ and $\Delta f_M=2$ Hz. From (7), $P_{Area,k}\le d_{\max}=(2\times150)/(0.02\times60)=250$ (MW). Let $P_{Area,1}=P_{Area,2}=P_{Area,3}=250$ MW and $d=d_1=d_2=120$ MW. The tasks for three subnetworks are described as follows.

Table 4: Areas in subnetwork 1

Area types	SN	Original nodes	Weight sum (MW)
CG areas	1	88–97, 101, 102	135.7
	2	47, 69	461.8
	3	79–81, 98, 99	219.8
	4	1–6, 8–11	135.2
	5	65	377.3
	6	26	303.0
	7	25, 27–29, 32, 113–115	5.3
	8	43–45, 48, 49	−115.2
Load areas	9	82–85	−130.0
	10	76–78, 118	−233.0
	11	68, 116	−184.0
	12	70–75	−199.0
	13	30	0.0
	14	13–17	−174.0
	15	38	0.0
	16	33, 34–37, 39	−173.0
	17	18–24	−167.0
	18	40–42	−199.0
NCG areas	19	100	−20.5
	20	7, 12, 117	−2.3
	21	86, 87	3.9
	22	46	−9.0
	23	31	−36.0

4.2.1 Subnetwork 1:

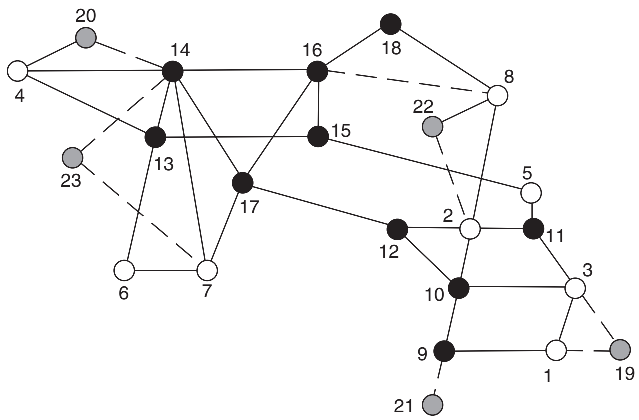
In $G_{SN,1}$ let generators 10, 25, 26, 49, 65, 69, 80, 89 be CGs and generators 12, 31 46, 87 and 100 be NCGs. Set CG, NCG and load areas as shown in Table 4. The weight sum of each load area is not more than 233 MW. The reduced graph $G^r_{SN,1}$ is given in Fig. 5*a*, where white dots are CG nodes, black dots are load nodes, grey dots are NCG nodes, and broken lines denote the edges that are removed by 'reducing irrelevant nodes and edges' and hence are not considered in phase-2. Table 4 and Fig. 5*a* renumber all nodes of $G^r_{SN,1}$ by the approach in [1]. New serial numbers are used in building OBDDs. From (9)

$$CSC_1 = \prod_{\forall i,j\in I_1, i<j}^{\otimes}\left[\overline{(A^*_{G,1})_{ij}}\oplus b_{1,i,j}\right]$$

4.2.2 Subnetwork 2:

In $G_{SN,2}$ let generators 54 and 65 be NCGs and the others be CGs. Set CG, NCG and load areas as shown in Table 5. Then $G^r_{SN,2}$ is given in Fig. 5*b*. From (9)

$$CSC_2 = \prod_{\forall i,j\in I_2, i<j}^{\otimes}\left[\overline{(A^*_{G,2})_{ij}}\oplus b_{2,i,j}\right]$$



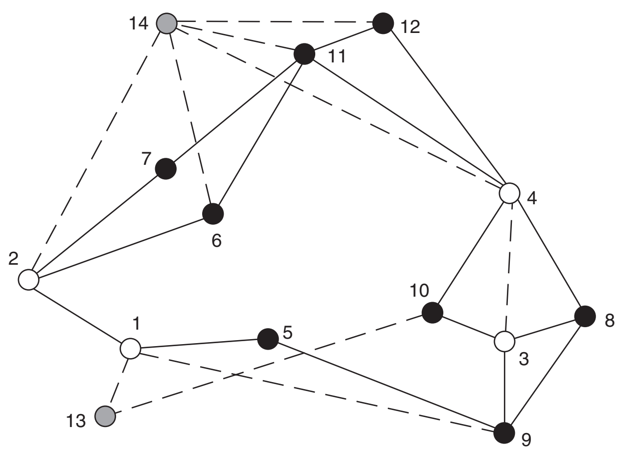


Fig. 5 *Reduced graph-models of subnetworks 1 and 2*
a Subnetwork 1
b Subnetwork 2

Table 5: Areas in subnetwork 2

Area types	SN	Original nodes	Weight sum (MW)
CG areas	1	66	340.4
	2	49	121.3
	3	61	154.3
	4	59, 63	−122.0
Load areas	5	67	−28.0
	6	51–53, 58	−70.0
	7	50, 57	−29.0
	8	60	−78.0
	9	62	−77.0
	10	64	−0.0
	11	56	−84.0
	12	55	−63.0
NCG areas	13	65	0.0
	14	54	−65.0

4.2.3 Subnetwork 3:

Since $G_{SN,3}$ has only 11 nodes, and generators 103 and 111 have much smaller generation capacities than generator 100, let all the three generators be NCGs and distribute loads to them to form three NCG areas as shown in Table 6. When they become asynchronous, some of lines 100-103, 103-104, 103-105, 103-110 and 109-110 are selected to directly form cut-set splitting strategies of $G_{SN,3}$. Then some OBDDs about $G_{SN,1}$ and $G_{SN,2}$ can be built in *offline layer*, e.g. $D(CSC_1)$, $D[(\boldsymbol{A}^*_{G,1})_{i,j}]$ $(i \in I_{G,1},\ j \in I_1)$, $D(CSC_2)$ and $D[(\boldsymbol{A}^*_{G,2})_{i,j}]$ $(i \in I_{G,2},\ j \in I_2)$. Their time costs are listed in Table 7.

Table 6: Areas in subnetwork 3

Area types	SN	Original nodes	Weight sum (MW)
NCG areas	1	100, 104–109	55.9
	2	103	16.4
	3	110–112	−72.3

4.3 Online layer

Node weights may be updated online according to new generation and load data. Then $D(PBC_1)$ and $D(PBC_2)$ are built online, whose time costs are also listed in Table 7.

4.4 Real-time layer

After asynchronous groups are detected, $D(SSC_1)$ and $D(SSC_2)$ are built at once. Then two final OBDDs, $D(CSC_1 \otimes PBC_1 \otimes SSC_1)$ and $D(CSC_2 \otimes PBC_2 \otimes SSC_2)$ are constructed from all available OBDDs. OBDD-based algorithms will find all cut-set splitting strategies satisfying PBC and SSC in the search space. Then a feasible splitting strategy will be given in phase-3.

This process is simulated by a typical case. Set two successive faults: at time $t = 0.0$ s a three-phase fault occurs near bus 100 at line 100-103; then another three-phase fault occurs near bus 80 at line 77-80 after 0.05 s. The two faults are both cleared at $t = 0.2$ s after local relays trip the two lines. As shown in Fig. 6, loss of synchronism and voltage collapse occur, and generators separate into four asynchronous groups {10, 12, 25, 26, 31, 46, 49, 54, 59, 61, 65, 66, 69}, {80, 89}, {100} and {87, 103, 111} within a short time. Once they are detected, *real-time layer* tasks are executed.

In phase-2 we have $I_{G,1} = \{1, 2, 3, 4, 5, 6, 7, 8\}$, $N_{G,1} = 2$, $I_{G,1,1} = \{1, 3\}$ and $I_{G,1,2} = \{2, 4, 5, 6, 7, 8\}$. Let $i_{G,1,1} = 1$ and $i_{G,1,2} = 2$. Since $G_{SN,2}$'s generators are synchronous, $I_{G,2} = \{1, 2, 3, 4\}$, $N_{G,2} = 1$ and $I_{G,2,1} = I_{G,2}$. Let $i_{G,2,1} = 1$. From (12)

$$SSC_1 = \prod_{\forall i \in I_{G,1,1}}^{\otimes} (\boldsymbol{A}^*_{G,1})_{i,1} \otimes \prod_{\forall j \in I_{G,1,2}}^{\otimes} (\boldsymbol{A}^*_{G,1})_{j,2} \otimes \prod_{\forall l \in I_1 - I_{G,1}}^{\otimes} \left[(\boldsymbol{A}^*_{G,1})_{l,1} \overline{\oplus} \left(\boldsymbol{A}^*_{G,1} \right)_{l,2} \right],$$

$$SSC_2 = \prod_{\forall l \in I_2}^{\otimes} (\boldsymbol{A}^*_{G,2})_{i,1}$$

After related OBDDs are built, OBDD-based algorithms can quickly find the cut-set splitting strategies satisfying SSC and PBC in $G^r_{SN,1}$ and $G^r_{SN,2}$. In sub-network 3, the only cut-set splitting strategy satisfying SSC and PBC is {100-103, 103, 104, 103-105, 109-110}, which can be given

Table 7: Simulation times of all tasks in three time-layers

Layers	Tasks	Simulation times (s)	
		Each task	Each layer
Offline layer	Build $D[(\boldsymbol{A}^*_{G,1})_{i,j}]$ $(i \in I_{G,1},\ j \in I_1)$ and $D(CSC_1)$	4.61	4.66
	Build $D[(\boldsymbol{A}^*_{G,2})_{i,j}]$ $(i \in I_{G,2},\ j \in I_2)$ and $D(CSC_2)$	0.05	
Online layer	Build $D(PBC_1)$	16.1	16.1
	Build $D(PBC_2)$	0.1	
Real-time layer	Build $D(SSC_1)$ and find cut-set splitting strategies satisfying SSC and PBC in $G^r_{SN,1}$	0.016	<0.2
	Build $D(SSC_2)$ and find cut-set splitting strategies satisfying SSC and PBC in $G^r_{SN,2}$	<0.001	
	Calculate an $\boldsymbol{A}^*$ $(S_{C,i})$	0.008	
	Check TVC for a splitting strategy	<0.0001	
	Check RLC for a splitting strategy	0.002	

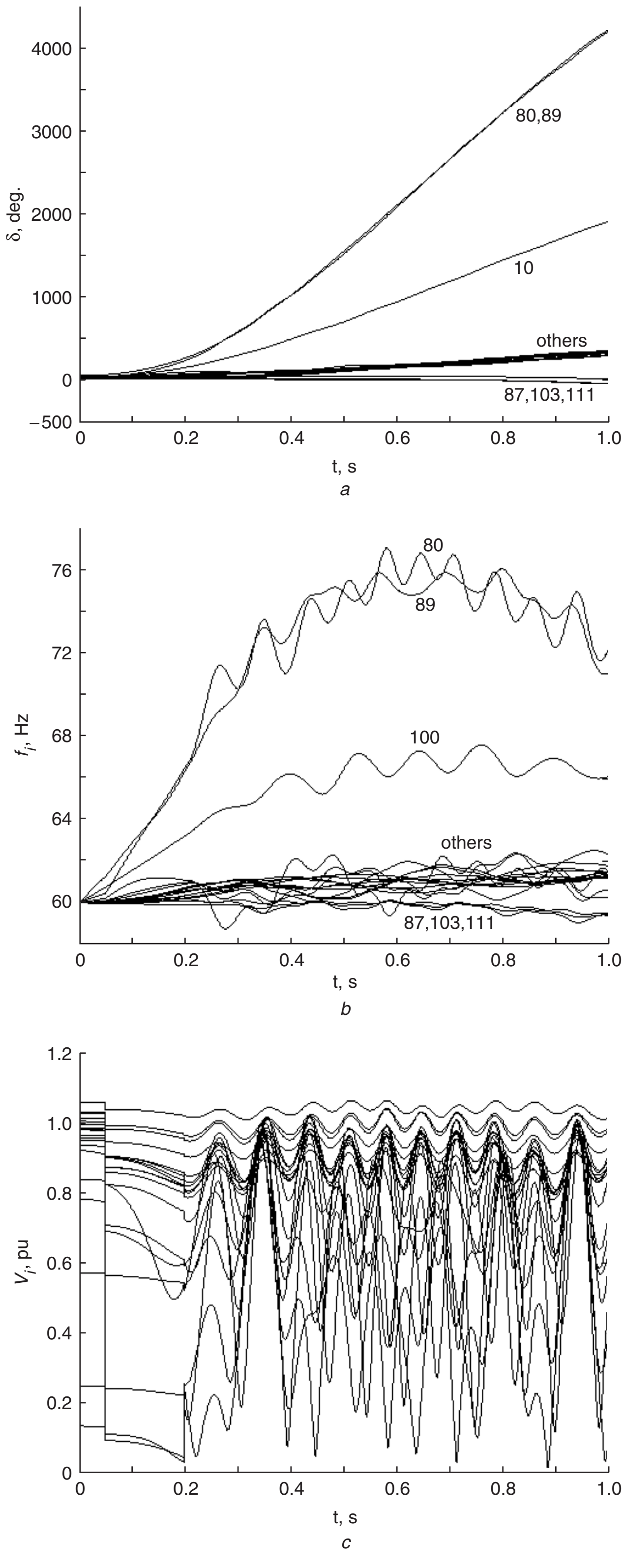


Fig. 6 *Dynamic curves of power network after faults are cleared*
a Angular rotor swings of all generators
b All generator frequencies
c Voltages of all generator buses

directly from Fig. 2 and Table 6. Consequently the cut-set splitting strategies of three subnetworks are incorporated to form the power network' cut-set splitting strategies. Finally, only two of them (denoted by $S_{C,1}$ and $S_{C,2}$) are found satisfying PBC and SSC in phase-2.

In phase-3, $S_{C,1}$ and $S_{C,2}$ are found satisfying both TVC and RLC, so they are feasible splitting strategies.

Table 8: Two feasible splitting strategies

	Feasible splitting strategies	Generators in each Island	Weight sum (MW)
$S_{C,1}$	69-77, 68-81, 75-77, 75-118, 77-80, 85-86, 92-100, 94-100, 98-100, 99-100, 100-101, 100-103, 103-104, 103-105, 109-110 (total 15 lines)	80, 89	−7.5
		100	35.4
		103, 111	−55.9
		87	3.9
		others	24.1
$S_{C,2}$	65-68, 68-69, 77-80, 77-82, 78-79, 85-86, 92-100, 94-100, 98-100, 99-100, 100-101, 100-103, 103-104, 103-105, 109-110 (total 15 lines)	80, 89	41.5
		100	35.4
		103, 111	−55.9
		87	3.9
		others	−24.9

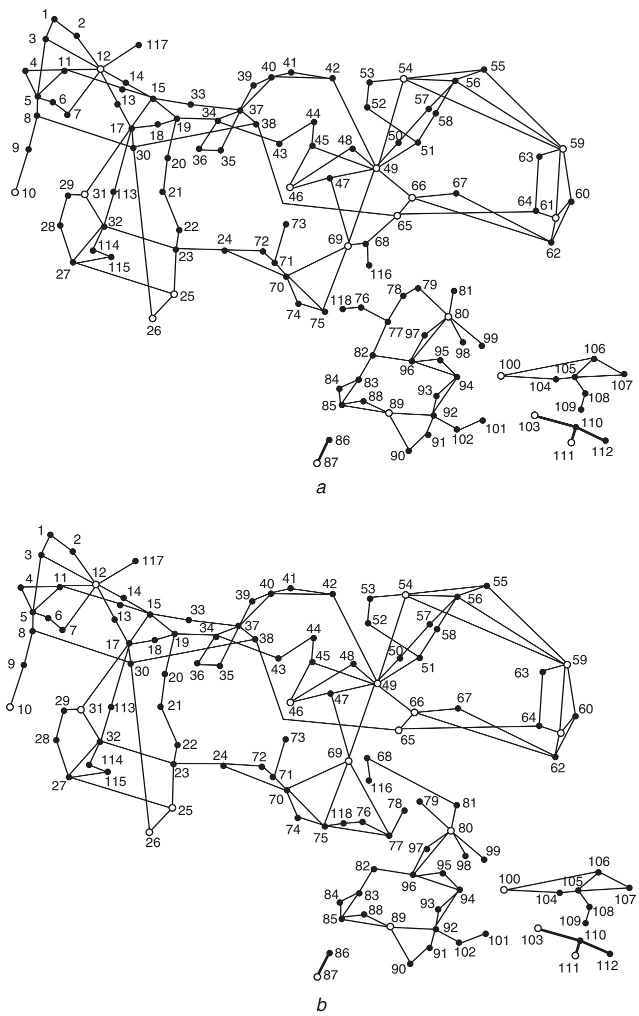


Fig. 7 *Power network after it is split by $S_{C,1}$ and $S_{C,2}$*
a $S_{C,1}$ (five islands)
b $S_{C,2}$ (five islands)

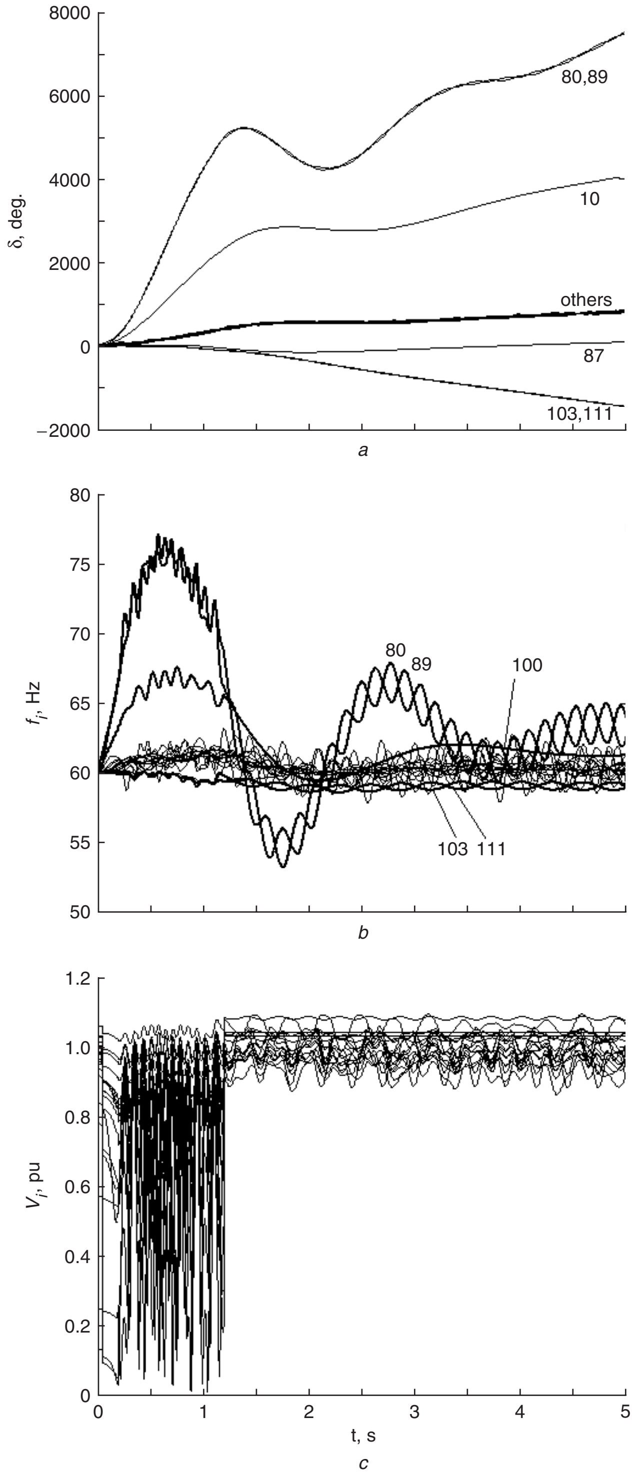


Fig. 8 *Dynamic responses of power network split by $S_{C,1}$*
a Angular rotor swings of generators
b All generator frequencies
c Voltages of generator buses

The online search time for them is less than 0.2 s. They are shown in Table 8 and Fig. 7, where lines in different islands are distinguished by different thicknesses. Each strategy produces five islands, which are all found stable by transient stability simulations. Figure 8 gives dynamic responses of the system after it is split by $S_{C,1}$ at $t = 1.2$ s (1 s after faults are cleared). Obviously system splitting makes all generators quickly stabilised.

Finally, the simulation time of each task in the three layers are given in Table 7, which considers the effects of parallel processing and neglects the communication time among processors. Although different cases may have different situations e.g. different numbers of cut-set splitting strategies, all *real-time layer* tasks can generally be finished within 1 s. Thus the modified method is able to find a feasible splitting strategy in real time.

5 Conclusions

Following the original method given in [1] for the splitting strategies ensuring steady-state constraints, and the simulation study on these splitting strategies in [3], we have proposed a modified method to real-time find feasible splitting strategies which can produce maintainable and stable islands. Compared with the original method the modified method is more efficient and practical for large power networks. Its performance has been shown by simulations on the IEEE 118-bus system.

6 References

1 Sun, K., Zheng, D.-Z., and Lu, Q.: 'Splitting strategies for islanding operation of large-scale power systems using OBDD-based methods', *IEEE Trans. Power Syst.*, 2003, **18**, pp. 912–923
2 Zhao, Q.C., Sun, K., and Zheng, D.-Z. *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.*, 2003, **18**, pp. 1556–1565
3 Sun, K., Zheng, D.-Z., and Lu, Q.: 'A simulation study of OBDD-based proper splitting strategies for power systems under consideration of transient stability', *IEEE Trans. Power Syst.*, 2005, **20**, (1), pp. 389–399
4 Vittal, V., Kliemann, W., Ni, Y.-X., Chapman, D.G., Silk, A.D., and Sobajic, D.J.: 'Determination of generator groupings for a islanding scheme in the Manitoba Hydro system using the method of normal forms', *IEEE Trans. Power Syst.*, 1998, **13**, pp. 1345–1351
5 You, H., Vittal, V., and Yang, Z.: 'Self-healing in power systems: An approach using islanding and rate of frequency decline based load shedding', *IEEE Trans. Power Syst.*, 2003, **18**, pp. 174–181
6 You, H.B., Vittal, V., and Wang, X.M.: 'Slow coherency-based islanding', *IEEE Trans. Power Syst.*, 2004, **19**, pp. 483–491
7 Ahmed, S.S., Sarker, N.C., and Khairuddin, A.B. *et al.*: 'A scheme for controlled islanding to prevent subsequent blackout', *IEEE Trans. Power Syst.*, 2003, **18**, pp. 136–143
8 Bryant, R.E.: 'Graph-based algorithms for boolean function manipulation', *IEEE Trans. Comput.*, 1986, **C-35**, pp. 677–691
9 Chang, C.S., Lu, L.R., and Wen, F.S.: 'Power system network partitioning using tabu search', *Electr. Power Syst. Res.*, 1999, **49**, pp. 55–61