A General Power System Network Topology Analysis Package and Its Applications

Chengming **Gong***, Shuhai Feng*, Xianfeng Xu*,Yuefeng **Liang***,Hongwei **Du***

Abstract – Power system network topology analysis is a basic function in the Energy Management System (EMS). The task of traditional topology analysis function is mainly to build bus and island models for online network analysis modules, such as state estimation, contingency analysis and dispatcher power flow. In this paper, some new topological objects in the power system network, including bay, loop, route, etc, and model type of node, bus, extended-bus, primary bus and primary extended-bus are abstracted. A power system topology analysis (PSNTP) package is addressed based on the abstracted concepts. The package is implemented in a general way by applying several basic graph theory algorithms to different model types. Functions and interfaces provided by the package are listed and some ways to improve the package's flexibility and efficiency are discussed in the paper. Application examples in Supervisor Control And Data Acquisition (SCADA) and state estimation modules show the package's validity and value.

Keywords: EMS, graph theory, Topology analysis.

1. Introduction

Most real-time application software modules in EMS are oriented to the node-model in which the detailed configuration, including breakers, disconnectors and bus bars, of plants and substations are concerned, while the internal algorithms of the software modules often base on the bus-model in which only impedance branches such as lines, transformers and series-wound compensators are reserved and groups of nodes connected by closed breakers and disconnectors are expressed as buses. Thus the connectivity of the node-model should be processed to build bus-model. The traditional topology analysis function accomplishes this task. Much work has been done on this basic function [1]-[3].

However, the function of topology analysis should not be limited to the analysis of bus and island. Much other topological information besides bus and island may facilitate or enhance special functions greatly. For example, in some state estimators, breakers are reserved as branches for the purpose of status errors identification, and a reliable by-pass identification algorithm addressed in [4] is based on the topological concept of bay. Loop and route are also useful concepts for analyzing or scheduling some special operation modes. Although different application may handle these topology analysis problems itself, it's desired to summarize the demands for topology analysis in the EMS and develop a general network topology analysis package. That many functions in the general package may share some core algorithms guarantees the stability and efficiency.

This paper abstracts some new topological objects and five model types in power system network and addresses a general power system network topology analysis (PSNTP) package based on these abstractions. The addressed package has been integrated in a real EMS product, serving as a shared library. More than five application modules have taken advantages of the PSNTP package, with old functions improved or new functions introduced. Some of the applications will be introduces in the following sections.

2. Abstraction and identification of some useful topological objects

Although different applications have different kinds of demands for topology analysis, some basic topological objects may play common roles in them. It's important to abstract these objects when designing a general-purpose topology analysis package. Eight topological concepts have been used in the addressed PSNTP package. They are extended-bus, bay, loop, route, breaker-terminal control relationship as well as the basic concepts in the traditional topology analysis function - node, bus and island. This section will describe these abstracted objects and discuss

* Nanjing Automation Research Institute. (gongchm@naritech.cn)

the algorithms for identifying them.

2.1 Node, bus and island

As is mentioned above, these three concepts are the fundamental concepts in power network topology analysis functions. A node is the minimum unit of connection points in the network. A bus represents a group of nodes connected by closed breakers and disconnectors, and buses connected by impedance branches compose an island. Commonly, basic Width-First-searching (WSF) or Depth-first-searching (DSF) algorithm in graph theory [5] is applied to identify bus and island objects with two steps: firstly, graphs with nodes as vertexes and breakers and disconnectors as edges are searched, and subsequently graphs with buses as vertexes and impedance branches as vertexes are searched. While the compositions of the graphs are different in the two steps, the search algorithm is the same. Noticing that the statuses of breakers and disconnectors change in sequence is helpful to improve the search efficiency.

2.2 Extended-bus

Breakers are modeled as branches in many applications. In these conditions, it's natural to abstract an object to represent the group of nodes connected by disconnectors. We name the object as extended-bus here. With this abstraction, a graph with extended-bus as vertexes and breakers and impedance branches as edges is used for further analysis. Same algorithm for identifying the bus and island may be applied to the extended-bus.

2.3 Breaker-terminal control-relationship

In the graph corresponding to the extended-bus, the so-called breaker-terminal control-relationship is abstracted to give the result which breakers control a given terminal of devices. This concept is useful at least in the following two conditions: 1) to assign measurements from breakers to devices terminals; 2) to select breakers to set outage of a device. To get the control breakers for a given terminal, the algorithm simply gets all breakers connected to the extended-bus of the terminal node.

2.4 Bay

A bay represents a set of connected breakers and/or disconnectors with physical bus bars as boundaries. This abstraction comes from the definition of bay in the widely accepted IEC61970-CIM standard [6], but only the topological aspects are reserved here. Identification of bays and their characteristics is helpful to arrange switch order automatically. The bay-identification algorithm bases on the basic search algorithm, but nodes of bus bars are marked as visited nodes, resulting in that the searching stops at these points.

Based on the abstraction of bay and breaker-terminal control-relationship, ref [4] addressed an algorithm which can identify the bypass operation mode reliably.

2.5 Route and loop

These two concepts are understandable from their literal meanings. It's a usual demand for finding the connectivity of two arbitrary points in the network and getting the routes if the points are connected. This is the demand for route searching. WSF algorithm is used to search whether route exists between two vertexes and the shortest one if route or routes found. If there are two or more routes exist between two points, any pair of routes composes a loop. It's also meaningful to give the whole information of looping in a network. The minimum spanned tree may be used to denote the whole looping information of the network because any link branch added to the tree will form a new loop. Loop and route are useful concepts for analyzing or scheduling some special operation modes.

Loop and route here have general meanings. In different conditions, the branches in route and loop may refer to different types of objects. Sometimes, the search may base on node-model, while in other conditions, the search may base on bus-model. Moreover, in different conditions, the status of breakers and disconnectors may be concerned or not. To adapt the algorithms for different demands, several model types will be abstracted in the next section. The route and loop concepts may apply to any of the model types.

3. Abstraction of Model Types

While discussing the abstraction of topological concepts in the above section, it has been pointed repeatedly that different kinds of branches are concerned in different applications. The extended-bus concept is abstracted to present the condition where breakers must be concerned as branches. Routine and loop must be adaptable for different model types. This section will give the abstraction of five model types classified by which kinds of branches types are concerned and whether switch status are concerned.

3.1 Model types differentiated by branches types.

Three model types are abstracted to presents models in which different branches are concerned. 1) Node-model indicates that all breakers and disconnectors are reserved as branches; 2) Bus-model indicates that only impedance branches are reserved; 3) Extended-bus model indicates that breakers are reserved as branches together with impedance branches. Statuses of breakers and disconnectors are concerned in the bus-model and extended-bus-model. All open switches are removed during searching.

3.2 Model types in which switch-statuses should be concerned

Two so-called primary models are abstracted to represent the models in which switch statuses are neglected. It's easy to prove that the primary model has the least set of buses or extended-buses.

Summarily, five model types are abstracted. Many search interfaces in the addressed package have a parameter specifying model type, thus different topology analysis demands are met in a general way by applying same basic algorithms to different model types.

4. Functions and interfaces

Based on the objects and model types described in the former two sections, a series of functions are provided in the package, most of witch will bill be introduces below in four type. Interfaces in C++ format of some functions will be given to further explain the functions if necessary.

4.1 Interfaces for creating model and setting operation mode

Original information of the power network is set during these interfaces. The model creation interface gets the information through a predefined structure, which contains identities and terminal node-names of all devices. Besides the branches, injections and ground disconnectors are also contained for identifying the energized or grounded status of an island. Different interfaces are provided to set operation modes. Callers may select one of them to set the status of one, more or all breakers, disconnectors or ground disconnectors.

4.2 Functions providing static topological information

These functions give the topological results independent to operation mode. Basically, bay information is provided by the following interface:

int GetBayInfo(BayResult_T &, const NodeName_T) ;

where BayResult_T is a pre-defined structure that contains the breakers and disconnectors in a bay.

Some other interfaces provide further results based on the analysis of bay characteristics, such as:

bool IsBypassBar(const DevId_T); and

bool IsBusLinker(const DevId_T);

which provide the characteristics of a bus bar or a breaker respectively.

4.3 Functions providing dynamic information

These functions give the topological results which may be affected by switch statuses. Results provided include bus, island, extended-bus, active breaker-terminal control-relationship and bypassed terminals.

4.4 Advanced functions

Although the above functions can give sufficient results for normal purpose, to provide some advanced search functions will facilitate the package-users. Three advanced search functions are provided in the PSNTP package. The interface pro-types are given and explained below:

int GetConnection(vector<DevId_T>&, vector<DevId_T>&, const NodeName_T&, const GraphModelType);

The first two parameters in this interface provide the injections and branches connected to the node specified by the third parameter.

int SearchRoute(vector<DevId_T> &, const pair<NodeName_T, NodeName_T>&, const GraphModelType);

The first parameter provides the branches in the route between the nodes specified by the second parameter.

int GetTrees(vector<DevId_T>&, const GraphModelType);

This function will return all the tree branches in the network in the first parameter.

All of these advanced functions are adapted for different model types, which are specified by the last parameter of each interface.

5. Implementation issues

The flexibility and efficiency should be concerned for industrial software. This section will discuss the Object-Oriented (OO) methods used to obtain the flexibility and some ways to improve the efficiency of the PSNTP package.

5.1 Detaching of interfaces and implementation

As the PSNTP package is implemented with C++ programming language, it's very convenient to adopt the OO properties of the language, the most important of which is to detach the interface and implementation of the package [7]. A virtual interfaces class is defined to describe the interfaces, and an implementation class is derived from the interface class. The implementation class is invisible to the users and the users' codes needn't be recompiled when the implementation class is modified.

5.2 Package layers

All the algorithms in the package are essentially based on the basic WFS or BFS algorithm from the viewpoint of gr aph theory [5]. The core graph theory algorithms are set as t he lowest layer in the package. The highest layer of the pac kage is the interfaces described in section 4.

An intermediate layer is set between the user interface layer and the core layer to manipulate the node names. Array is the data structure used in the core layer, but the node names and device identities in the user interfaces layer may be discrete values and even strings. Index should be created to carry out the conversion between subscripts and names. We ever set the index map in the core algorithm, but we decided to find new ways after observing that the frequent conversion of name and subscript in the core algorithm became the main consumer of CPU resources when the package was applied to large-scale network. Finding that the conversion may be carried out in batches, we set the intermediate layer in the package. Then the index map is created while creating the graph for the core layer and will be used for converting the results in subscript form to names. The efficiency of searching is improved about ten times after the intermediate layer is set.

6. Application examples

Some application examples such by-pass identification has been mentioned above. In this section, two more applications of the package in operation simulation and state estimation modules will be given.

6.1 PSNTP applied in operation simulation

It's meaningful to simulate operations in the EMS before actual remote-control taken. There are many rules that constraint the operations, but most of the rules, from avoiding to connect the energized and grounded islands, to preventing outage or run devices by disconnectors, may be judged by topology analysis. We have developed an operation simulation (OPSIM) package based on the PSNTP package. In the OPSIM package, bus and island information from the PSNTP is used to avoid connectting energized and grounded islands. Furthermore the *GetConnection()* interface supports the OPSIM to give the switch orders when the operators prepare to run an outage device.

6.2 PSNTP applied in parameter error pre-processor

Modern state estimators can identify and estimate the parameter errors [8]. If a pre-process select the suspicious branches for further processing, the efficiency of the whole identification of parameter errors will be improved. We developed a pre-processor which base on the criteria that the sum of the production of real power and reactance of each branch in a loop is close to zero. It's necessary to find all the independent loops in the network to implement this pre-processor. The *GetTrees()* interface in the PSNTP package just suits for this case.

7. Conclusion

Some work relating to enhance the topology analysis function has been introduced in this paper. The abstraction of some new topological concepts and model types, which makes the basis of the addressed PSNTP package, is discussed. Although the package has greatly benefited some other real-time functions, there may be more useful and interesting objects to be abstracted. It's worthy to pay more attention on the enhancements of topology analysis function, and this kind of work may be carried out more efficiently if both the topological characteristics of power sytem network and the different kinds of theory graph algorithms are comprehended.

References

[1] Erkeng Yu, "Energy Management System (Chinese)," Beijing: Science Press, 1998.

[2] Phongsak D. Yehsakul, hj Dabbaghchi, "A Topology-Based Algorithm For Tracking Network Connectivity," *IEEE Trans. Power Systems*, Vol. 10, No. 1, pp. 339-346, Feb. 1995.

[3] Ying He, David C. Yu, Youman Deng, Jiansheng Lei. "An Efficient Topology Processor for Distribution Systems", in *Proc. 2001 IEEE PES Winter Meeting*, pp. 824-829.

[4] Chengming Gong, Yijun Yu, Xiaopeng Liang, "The Algorithm of Discernment of Bypass (Chinese)," *Jiangsu Electrical Engineering*, Vol. 24, No.4, pp. 10-12, Jul. 2004.

[5] Robert Sedgewick, "Algorithms in C++, Part 5: Graph Algorithms," 3rd ed., Boston: Pearson Education, 2001.

[6] IEC, "Draft IEC 61970 Energy Management System Application Program Interface (EMS -API), CCAP Guideline Preliminary Draft," 1999.

[7] Bjarne Stroustrup, “The C++ Programming Language,” Special 3rd ed., Indianapolis: Addison-Wesley Professional, 2000.

[8] W.-H. Edwin Liu, Swee-Lian Lim, "Parameter error identification and estimation in power system state estimation, " *IEEE Trans. Power systems,* Vol. 10, No. 1, pp. 200-209, Feb. 1995.