A Graph Based Method for Faster Display for Distribution Networks

MANSOOR AL-A'ALI
Computer Science Department
University of Bahrain
PO Box 320328
BAHRAIN
mansoor.alaali@gmail.com malaali@itc.uob.bh

Abstract: - This paper presents our logical display approach for Distribution Networks based on the concept of Microsoft graphs. We demonstrate that by using Microsoft graphs we do not have to redraw the circuits after each switching operation or after a change in the status of the circuit. This work is a follow up on our previously published research on the concept of logical display algorithms which greatly enhances the speed of switching and display operations as well as making the displayed circuit much more readable and traceable by the control engineers [1], [2] [3] & [20]. We demonstrate that by using Microsoft graph concept we can speed up the different switching and tracing operations by a significant percentage. This contributes to solving the current problems faced by control engineers in such a way to enable them to tend to network problems more promptly.

Key-Words: - Distribution, network systems, algorithms, logical display, Graphs

1 Distribution Networks

A distribution network is normally drawn or digitized using a graphics package such as AutoCAD or Microtation, see Fig. 1. Systems to process the required network operations are programmed to carry out many of the control centre operations such as adding new stations, retiring old stations, tracing the flow of a given circuit, isolating a station for maintenance, isolating a complete circuit for maintenance, redirecting the flow of electricity, etc. These operations require decision making capabilities which can be provided either by the engineer at the control centre or by the system. Speed of display of these schematic diagrams and the speed by which the system responds to the commands to carry out the switching operations and then redrawing the circuits correctly and showing the correct status of switches and circuits is of vital importance. The sample network shown in Fig. 1 was taken from a Distribution Control Centre which was drawn using Microstation and programmed by using Bently's MDL. The system suffered from two main problems: 1) a very large confusing schematic display comprising over ten thousand stations and 2) speed problems where one circuit may sometimes take up to three minutes to display. For each switching operation, the system has to redisplay a fixed schematic diagram consisting of tens of thousands of elements, each with coordinates, colour, thickness, status, etc. This has prompted us to search for a better technique for display which would be much faster than the current system. Our approach is based on a logical generated graphical display of the network where both the graphical information and the status information is all kept in the database in RAM and is extracted for the circuit concerned. The coloring scheme showing that a circuit is live or dead is also generated directly. This meant that we can deal with individual circuits at any one time, only components of the circuit concerned is generated. This approach proved to be much faster and more informative.

A control engineer normally works on more than one circuit at the same time where he communicates with the field engineers who carry out his instructions. On a large-scale system, each switching operation may take from few seconds to few minutes and this causes aggravation and loss of control. A typical network will have about 60 primaries, almost 9000 substations, and almost 200 switching stations, 1500 overhead poles, and 25000 switches. This is a huge undertaking for any multi-user client server environment with both database and graphics processing.

Research on automating the Electricity Distribution Control Centers and their activities of monitoring and controlling electricity distribution networks has received some attention in the research literature. Most of the research is focused on post fault reconfiguration [4], [13]. However, no research was directed to the issue of the speed of displaying the graphical distribution of the network which will enable the control engineer to visually study the situation and take the appropriate action. Most of the systems available are simple schematic displays.

Some studies were conducted on the issues relating to the problems speed efficiency of displaying distribution networks [1], [2] & [7]. After thoroughly searching the literature, we have found no other reported attempts to present some solutions to the issue of speed problems of distribution control systems. We have found no published work about the problems and methods of display of the distribution network. Clavijo etal [6] describe a distributed system based on CORBA technology to provide real-time visual feedback to operators of large supervision systems. Dash, etal [8] present a functional-link-neural network for short-term electric load forecasting. Tae-I Choi etal[15] report on a remote control and fault location system for distribution lines. Hua etal [11] propose an algorithm and implementation of an incident based connectivity trace system for distribution network. Carvalho etal [4] proposed an approach to operational planning and expansion planning of large-scale distribution systems.

Herrell etal [10] proposed a method of modeling of distribution systems in PCs. Yeh, etal [18] present an integrated solution for computerized distribution planning in a geographic information system (GIS) context, a synergy that magnifies the data accessibility between load forecasting and feeder planning tools, sealing the traditional gap between long-term and short-term distribution system planning. According to these authors, this approach enables the distribution system studies in a GIS context, to best assist utility planners in deciding where and when the customers will grow and how to expand the system facilities to meet the demand growth. Wainwright [16] present a Distribution Engineering Geographical Information System as a major program of network information data capture at all voltage levels from the customer service cable right through to the primary supply in-feed; a suite of software for viewing the data in the office and in the field; tools for network optimization; and a range of automation and information processing systems which assist in the operation and maintenance of the distribution network. Lestan etal [12] present a form of a distribution network automation utility experience.

Cavattoni etal [5] describe a GIS experience of a company that manages Verona's utilities networks. It outlines some particular aspects of the GIS project, of the numerical cartography and of the utilities network database. It also describes two utilizations of geographic information systems to handle data referring to geographical positions giving the length calculation of 50 electricity distribution lines. Xing Weiguo etal [17], report on a proposal for a PC-based distribution system without giving specific achievements or results.

2 Microsoft Graphs

Research on graphics is extensive and tackles this area from a number of different perspectives [20], [21], [22], [23]. A graph, like a tree, is a collection of nodes and edges, but has no rules dictating the connection among the nodes, see Fig. 2 & 3 from [14]. Realize that all trees are graphs. A tree is a special case of a graph, one whose nodes are all reachable from some starting node and one that has no cycles. Figure 3, taken from [14] shows three examples of graphs. Notice that graphs, unlike trees, can have sets of nodes that are disconnected from other sets of nodes. For example, graph (a) has two distinct, unconnected set of nodes. Graphs can also contain cycles. Graph (b) has several cycles. One such is the path from v1 to v2 to v4 and back to v1. Another one is from v1 to v2 to v3 to v5 to v4 and back to v1. (There are also cycles in graph (a). Graph (c) does not have any cycles, has one less edge than it does number of nodes, and all nodes are reachable. Therefore, it is a tree.

The edges of a graph provide the connections between one node and another. By default, an edge is assumed to be bidirectional. That is, if there exists an edge between nodes *v* and *u*, it is assumed that one can travel from *v* to *u* and from *u* to *v*. Graphs with bidirectional edges are said to be *undirected graphs*, because there is no implicit direction in their edges [14].

When drawing a graph, bidirectional edges are drawn as a straight line, as shown in Figure 3. Unidirectional edges are drawn as an arrow, showing the direction of the edge. Fig. 3 shows a directed graph where the nodes are Web pages for a particular Web site and a directed edge from *u* to *v* indicates that there is a hyperlink from Web page *u* to Web page *v*. Notice that both *u* links to *v* and *v* links to *u*, two arrows are used—one from *v* to *u* and another from *u* to *v*.

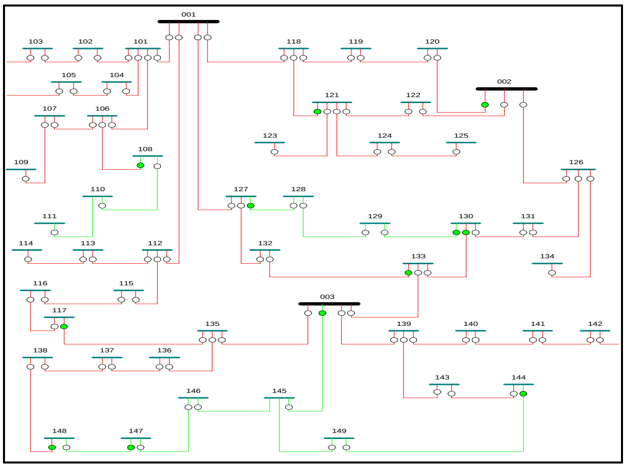

Figure 1: A sample of a manually digitized distribution network

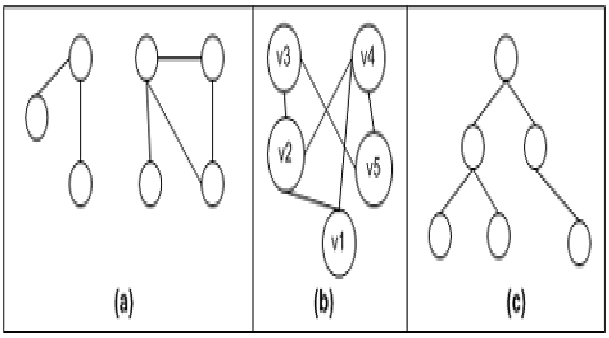

Fig. 2 a.Three examples of graphs

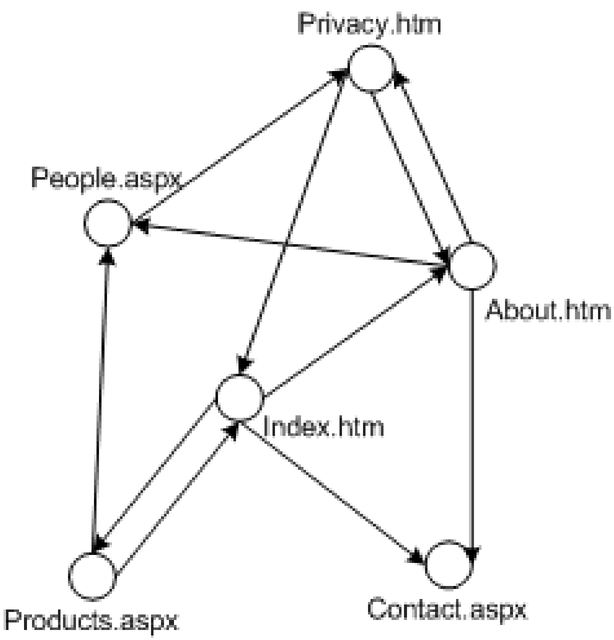

Fig. 2 b. Example of a directed graph

3 Different Techniques for Distribution Networks

A Distribution Network System is a collection of circuits consisting of primary stations (feeders), thousands of substations (distribution), hundreds of switching stations and thousands of overhead poles all linked via switches and cables, see Fig. 1 & 4. Overhead poles contain special switching devices such as jumpers and fuses. Stations are linked to each other through switches, which are fixed in the stations. Every station may contain one or more switches and has one or more bus-bars, where every bus-bar can be considered as a station by itself. Primary stations are the sources that feed other stations, except in some rare situations where they act as a switching device. Substations are the stations that are fed by primary stations. Switches on primary stations are called circuit breakers. All connecting stations linked to a circuit breaker represent an electrical circuit. Electrical circuits form the Electrical Distribution Network, see Fig. 4. The control engineer can either zoom in or zoom out on these circuits which normally makes the circuits either unreadable, in the case of zooming out, or only one part is readable in the case of zooming in. A sample of the network is shown in Fig. 1, where a primary source 001 feeds substation 101, 112, 127 & 118. Substation 101 has four switches, and it feeds substations: 102, 104, and 106. The sample network shows three primaries, 001, 002 & 003.

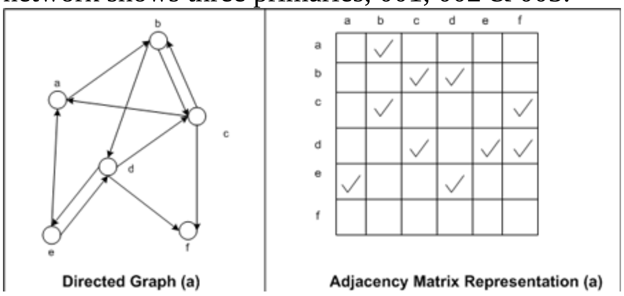

Fig. 3 Model of pages making up a website

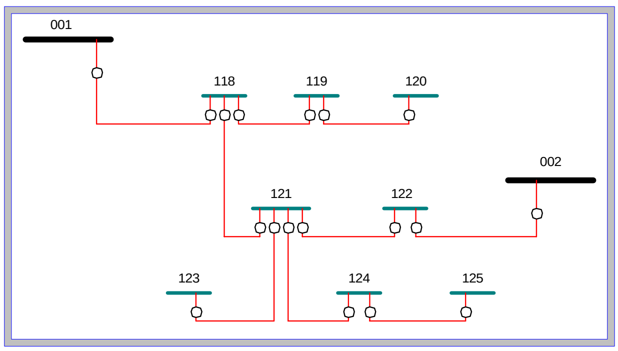

Fig. 4 A sample distribution network

These monitoring and control activities of a distribution control centre include switching the flow from 'on' to 'off' and vv, preparing switching plans, adding and removing substations, adding switches, adding and connecting cables, etc. Such activities have been explained in [1]. To operate in such environment the control engineers need to get quick, reliable, get up-to-date and detailed information about any circuit or device of this network system at any point in time.

In order to make the system as quick as possible, the elements of the network must be modeled in such a away that optimizes the time of update and the graphical display of new results. The structure of the database tables and other supporting data structures used in the programs form vital factors towards achieving this goal. In addition, special attention was given to the design of the algorithms which carry out all the required switching, tracing and other processing operations. The system presented in [1] was developed using Delphi and a DBMS whilst the new EDMS was developed using C# which is one of the .NET platform programming languages and fully object-oriented language and using Microsoft SQL Server 2005 as the external data storage. Both systems set out to perform the different network operations that deal with the distribution network. The EDMS presented in [1] design strategy was based on two types of data storage: 1) External Data Storage (the database schema) and 2) Internal Data Storage: Stack. The new Graph based EDMS is based on: 1) External Data Storage (the database schema) and Internal Data Storage (Stack and Graph). Both systems use the Stack as the internal data storage to perform the different network operations because the stack is the only data structure suitable for traversing distribution networks where each node may have one or more outputs and one or more inputs. The stack is used to show the connectivity of the network and hence the flow of power through the stations which would dynamically change if any of the switches is switched on or off. Physical connectivity can be represented using an ordinary data structure such as a database table.

Graphs are composed of a set of nodes and edges, just like trees, but with graphs there are no rules for the connections between nodes. With graphs, there is no concept of a root node, nor is there a concept of parents and children. Rather, a graph is a collection of interconnected nodes. Systems such as Electricity Distribution Management Systems (EDMS) can be modeled using graphs where stations can be modeled as nodes and cables connecting stations through switches as edges.

When the electrical network drawn and the system user has to perform tracing through the network or making switching plan, that is change the status of the switch from on to off or vise versa, we need such design that limits the access to the external database. For example, when the user wants to trace from the station number 118 as shown in Fig. 4, the system had to read the same data from the database and perform the same steps done by the drawing procedure except that the traced stations will be colored with a different color. If we calculate the time required to perform tracing in any network we find that it consumes much time, the time consumed on drawing and on changing the color of the traced stations. But when we use the graph technique, this time could be reduced 2 times since it will consume time on changing the color of the traced stations. Therefore, the main reason behind using graph is to perform the network operations dynamically and limits the database access for just updating the records. Each node in the graph will hold station records, while the edge will hold the cable and the connecting switches to that cable.

```
get starting station
push all out switches into Stack
set column  to 0
set level to 0
draw at  (column, level)
insert node to the graph
increment column
repeat
        pop switch
        get base station
        get column and level from xy-coordinates
by the mapping function which has been
defined earlier
        increment column
        counter = 0
        flag = True
        while flag
                get connecting station
                 add edge to base node
                if found then
                        increment counter
                        if out_switch = on then
                        get base station
                        else
                        flag = False
                else
                flag = False
        end of while
        flag = True
```

starting_point = (column, level)
look for "counter" empty adjacent cells in the grid moving horizontally from the starting point
if found then
flag = False
else
increment level
draw at (column, level)
insert node to the graph
until Stack empty

Fig. 5 Algorithm for logically drawing the distribution network

4 Using Graphs and Adjacency Matrix to Draw Distribution Networks

For each station in the electricity network there could be a maximum of five edges, representing the connection from a switch on that station to another switch on another station. Each station record is stored on a Node instance and the connection record (cable and the two switches) is stored on an Edge to Neighbor instance. And for each node there is a list of nodes representing the neighbors of that node (i.e. station). And if we use Fig. 1 as an example, the representation of that network will be as shown in Table 1.

List of (Node) stations	*Neighbors (stations related to that station)*
001	*118*
118	*119 - 121*
119	*120*
121	*122 – 123 – 124*
122	*002*
123	
124	*125*

Table 1 Network Representation using the Adjacency Matrix

Code: 001, Bus-bar Code: A and Switch Code: 3, the time required to draw the above network was 0.0968. Assume that we want to change the status of that switch (the switch defined by the user) from ON to OFF, we need to find out how much time is required by the system to perform such operation. Time required to change the color of the connections of the eleven stations in Fig. 6 as well as updating the database was 0.193. This time could be much higher without the help of the graph. In our case it could be 0.0968 (time required for drawing) + 0.193 (time required for updating the network elements record) = 0.2898 seconds.

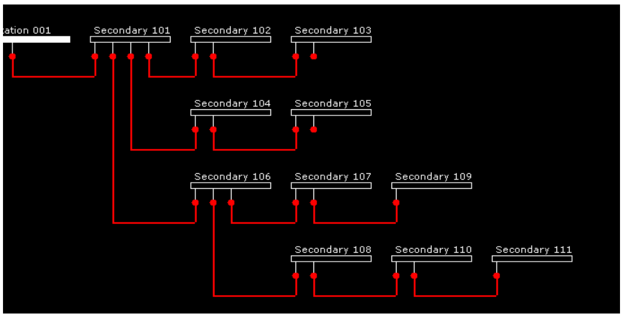

Fig. 6 a: Electrical Network using the new EDMS

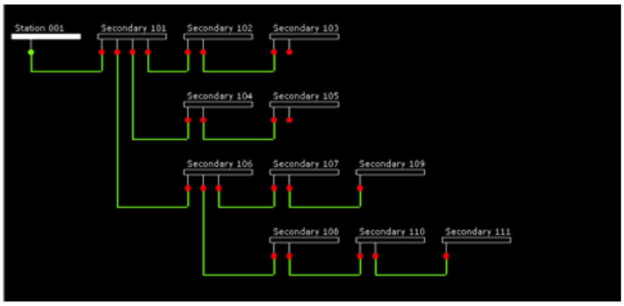

Fig. 6 b: Electrical Network using the new EDMS

5 Results

Since the original aim of the project was to devise a number of approaches to display circuits and compare between their outputs, we measured the results in terms of speed. The speed is measured in milliseconds and calculates the time interval between getting the source station from the user just after pressing the start button and drawing the last station on the diagram.

Six different circuits were tested and the results were collected and compared, see Table 2.

Table 2 Results by the Graph based approach

Number of Stations in circuit	*8*	*14*	*32*	*43*	*75*	*138*
Speed (s)	*0.074*	*0.136*	*0.264*	*0.308*	*0.592*	*1.442*

By switching we mean either turning a switch on or off. The efficiency of the algorithm will depend on the number of switches affected to the time spent to show the effect of this switching in the diagram. Here is a list of tests done to check the efficiency of the switching algorithm:

We show in Figs. 8, 9, 10 & 11 the comparison between the Graph based approach and the non-graph based approach for all of our four different display algorithms for the graph based and non-graph based approaches. We can see that the graph based approach which we term our algorithm sustained an improvement in speed between 200ms and 400ms throughout six different test circuits. The upper curve in Figs. 7, 8, 9, & 10 shows the non-graph based results and the lower curve shows the graph based results. We have to remember here that these in both algorithms are logical displays. The logical display approach was demonstrated to be ten times faster than the schematic display approach [1]. Fig. 11 shows the speeds of the Graph based algorithm and the non-Graph based for the four different algorithms

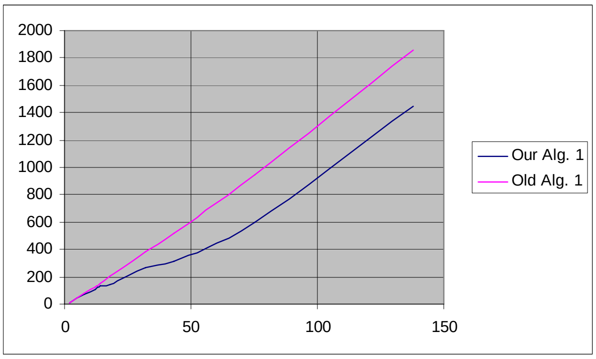

Fig. 7: Speeds of the Graph based and the non-Graph based algorithm 1

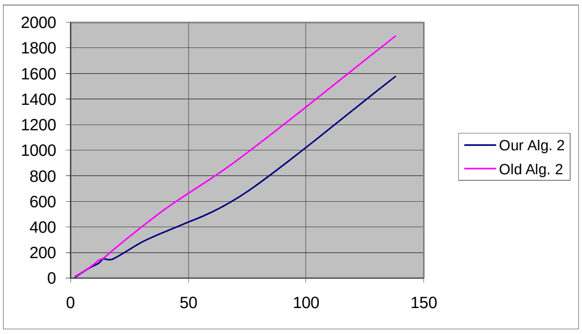

Fig. 8: Speeds of the Graph based algorithm and the non-Graph based algorithm 2

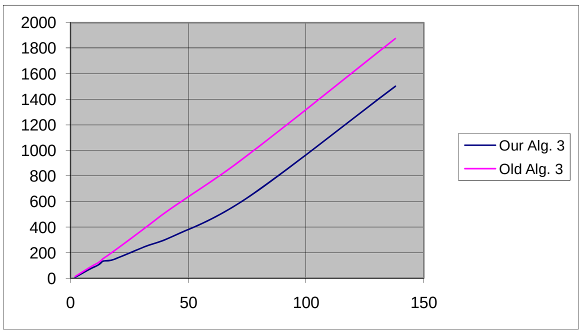

Fig. 9: Speeds of the Graph based algorithm and the non-Graph based algorithm 3

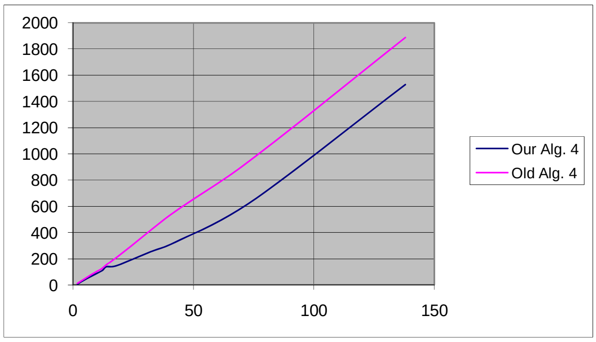

Fig. 10: Speeds of the Graph based algorithm and the non-Graph based algorithm 4

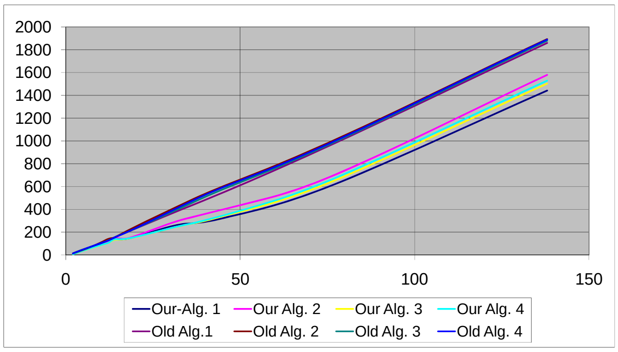

Fig. 11: Speeds of the Graph based algorithm and the non-Graph based for the four different algorithms

6 The Logical Approach

This section explains the procedures and operations, which are used to perform different network operations on the Electrical Distributed Management System. We present four different approaches designed to display the circuits in a logical manner, which reflects the structure of the network and the status of the circuits in the network.

The actual techniques of how the display is achieved and how elements on the circuits are made dynamic are not discussed as it is beyond the scope of this paper. Further, no description is given about the internal operations of the system or about the database structures necessary for the same reason.

Changing the status of switches on stations changes the flow of electricity and the status of the connecting circuits, which would require the display to be changed. Stations, cables, switches and bus-bars which are in 'off' position must be changed to the color green and back to red once electricity flows through again. The logical display allows actions to be taken on switches to change their status and hence the objects displayed are dynamic.

6.1 Draw Procedure

This procedure draws the complete circuit of the given station to the open points ('off' switches) or to the connecting primary station. Firstly, the starting station is taken as an argument. All the 'on' switches in this station are pushed in the stack. Then repeatedly an element of the stack is popped and the connecting station of the switch in that element is looked up. At this stage both stations are drawn by applying one of the algorithms. Then all of its 'out' ('on') switches are pushed into the stack, and so on. This operation of pushing on the stack continues until reaching an open point ('off' switch) or a primary station (circuit breaker). The procedure terminates when the stack is empty.

6.2 Trace Procedure

This procedure traces all the connecting stations of a given station/switch. It traces and highlights a given switch to all open points or a primary based on the existing network. It takes the same steps as the Draw procedure except that when a switch is popped, a special test is performed. This test checks if the switch is the desired one. If this is the case, the draw procedure is repeated using another stack.

6.3 Tracing through Open Points

This procedure will draw the complete circuit through a given open point until a primary is reached. All other open points after the given open point will be passed. The circuit starting from the specified open point to the primaries is highlighted. This procedure takes the same steps as the Draw procedure except that when a switch is popped a special test is performed. This test checks if the switch is the desired one. If this is the case, the draw procedure is repeated using another stack.

6.4 Grid

All the procedures that are used to carry out the various network operations on the network are performed logically on a grid and physically on a panel. The grid is defined as a two-dimensional array initially set to empty. Cells are marked as non-empty if they are occupied by an element on the panel. Two procedures are used to perform the mapping function between the grid cells and the panel pixels.

To draw an element on the panel, its position is marked first on the grid, and then the mapping function is used to get the corresponding x-y coordinates on the panel and finally draw the element at that position.

7 The four logical display approaches

7.1 First Approach

The user will identify the starting point of the circuit that is required to be displayed which is normally a switch, a bus-bar or a station. The first station is drawn in the top left cell. The next station is placed to its right. Then each time an element is popped (representing a station), it is placed to the right of the current station's position. At the same time, the variable that represents the vertical cells in the grid (column) keeps incrementing each time an element is popped and drawn. This will continue until an 'off' switch is encountered. At this stage, the variable that represents the horizontal rows in the grid (level) is incremented by 1, and the next station will be drawn at the (column, level) position.

The previous steps are summarized in initializing the Column and Level variables to 0 once only prior to commencement of drawing, incrementing Column each time a station is drawn and incrementing Level each time an 'off' switch is encountered. The two variables are never reinitialized or set to any value at any stage of the drawing phases.

7.2 Second Approach

The second algorithm is more intelligent than the first, since it does not keep incrementing the x and y axis each time a station is placed in a dull fashion, instead it sets the variable to a specific value at each step based on the algorithms' rules.

In this case, the level (row) is incremented in the same way as algorithm one, but differs form it when we move vertically down. Each time a switch is popped, its connecting station is pre-traced to determine the number of stations that can be placed on one line horizontally. In other words, when a switch is popped, its connecting station is searched for. Then only one switch getting out from the connecting station is pushed. This switch is popped and again its connecting station is found. These two steps are repeated until encountering an off switch or arriving at a station that has no out switches. Each time a switch is pushed, indicating that a new station is found, to be drawn on the same horizontal line, a counter is incremented. At the end of this short loop, we will end up with the total number of cells needed adjacent to each other. Then a function is called to scan the grid starting from the current station position, i.e. (column +1, level). Starting from this cell and moving to the right, this function will check whether the required number of cells is available or not. If not, level is incremented by one and the function is recalled. This loop will terminate whenever the function reaches the desired row and will end up drawing the stations, which were pre-traced from the starting station in that row. Accordingly the grid cells are updated. It can be observed from this approach that space utilization and readability has improved as circuits can be traced more easily and clearly.

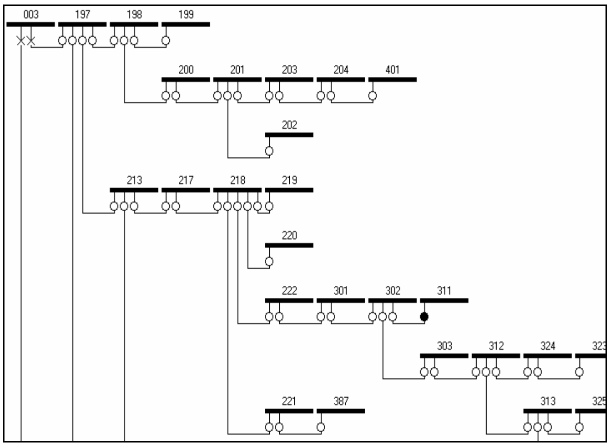

Fig. 12 Second approach example

7.3 Third Approach

In the second algorithm a large number of cells were skipped because of the nature of the algorithm, but the third approach provides a better space utilization technique. This algorithm does not perform any kind of pre-tracing mechanism. Its technique is that whenever it finds an empty cell starting from the first stations right adjacent cell and moving horizontally then vertically (never move backwards, i.e., decrementing the column or level), it draws the station directly with the need to pre-trace through the table.

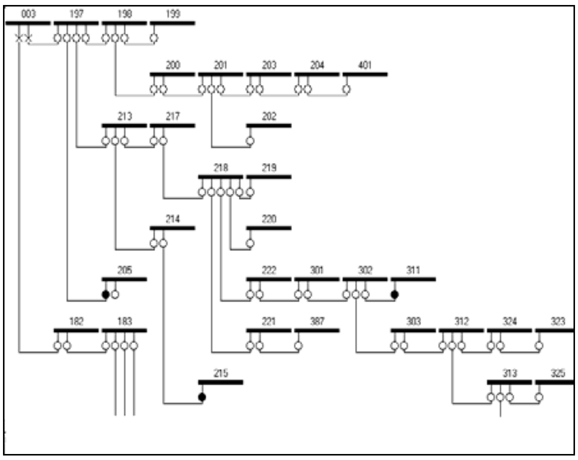

Fig. 14 Another third approach Example

Whenever a station is drawn, one of its out switches is popped and the corresponding station to be connected is searched for. If the adjacent cell is empty then st2 is placed there. If it is occupied with another element then we move one step vertically to check the next row. This procedure is repeated until an empty cell is found, in which case st2 will be placed on that cell. See Fig. 4. In this approach space utilization has greatly improved.

7.4 Fourth Approach

The three previous approaches work in the same manner due to the fact that we keep moving to the right and left only. The fourth approach differs in such a way that through scanning the diagram via the grid, we may need to move upwards looking for empty cells.

Let us assume that st1 is the first station and st2 is the second. Flow is from st1 to st2.

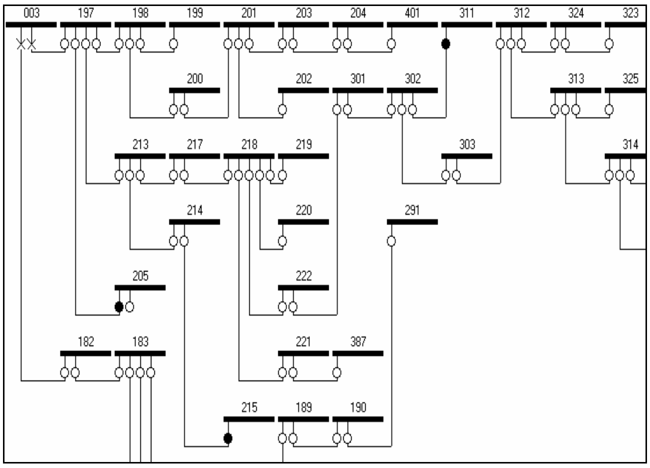

Fig. 15 Third approach Example

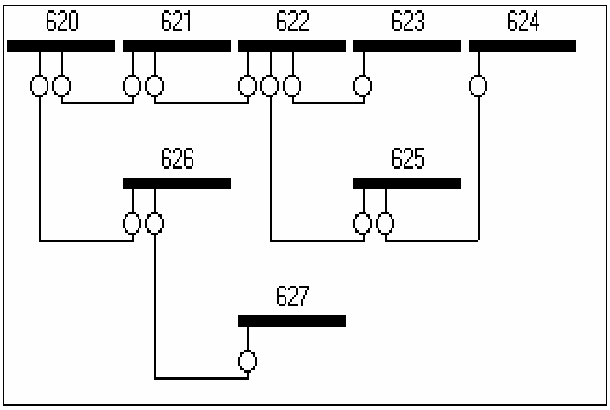

Fig. 16 Fourth approach Example

The logic is similar to the third algorithm, except that it does not directly draw st2 if the adjacent cell to st1 is empty; instead, it moves upward by decrementing the variable representing the row until reaching an occupied cell. At this stage st2 will be placed. This approach has improved space utilization drastically, but the readability has decreased, see Fig. 5.

8 Conclusion

This research is a continuation on our previous published research on the switching and display of electricity distribution network systems. Our previous work concentrated on developing new algorithms to logically display these networks which were demonstrated to speed up the switching and display as well as presenting logical display techniques which enable the control engineer to take immediate and informed decisions on different situations of the network especially in emergency situations. The research presented in this paper studied the utilization of Microsoft graphs to further improve the speed of display of the network circuits. The improvement in speed drawing and redrawing any circuit was improved by between 200ms and 400ms depending on the size of the circuit.

Reference:

1. Al-A'ali Mansoor a, "Four new efficient logical display techniques of electricity distribution networks", *Journal of circuits, systems & computers*, 2007, World Scientific Publishing Company.
2. Al-A'ali Mansoor b, Efficient Graph Algorithms for Logically Displaying Electricity Distribution Network Systems, MEST – Middle East Spacial Technology Conference, Bahrain Society of Engineers, Dec. 2007.
3. Al-A'ali Mansoor, "The Effect of Different Data Structures on Speed in an Electrical Distribution Network", *International Journal on Soft Computing*, vol. 5 (1), pp. 1-12, 2008.
4. Carvalho, P.M.S., Barruncho, L.M.F., "An evolutionary approach to operational planning and expansion planning of large-scale distribution systems", Ferreira, L.A.F.M., 1999 *IEEE Transmission and Distribution Conference*, 1999.
5. Carvalho P.M.S., FerreiraaL.A.F.M., and Barrunchob L.M.F., "Optimization approach to dynamic restoration of distribution systems", *International Journal of Electrical Power & Energy Systems* Available online 11 October 2006.
6. Cavattoni, E., Giusti, M., Rumor, M., McMillian, R., Ottens, H.F.L, "GIS for utilities networks: the Verona's experience", Geographical Information from Research to Application through Cooperation. Second Joint European Conference and Exhibition. IOS Press, Amsterdam, Netherlands. pp. 1123-32 vol.2, 1996.
7. Clavijo, J.A., Segarra, M.J., Baeza, C., Moreno, C.D., Sanz, R. , Jimenez, A., Vazquez, R. , Diaz, F.J. , Diez, A. "Real-time video for distributed control systems", *Control Engineering Practice*. Vol. 9, Iss. 4, pp. 459-66, April 2001.
8. Curcic, C. S. Özverena, L. Crowea and P. K. L. Lob, "Electric power distribution network restoration: a survey of papers and a review of the restoration problem", *Electric Power Systems Research*, Volume 35, Issue 2, November 1995, Pages 73-86
9. Dash, P.K., Liew, A.C., Satpathy, H.P., "A functional-link-neural network for short-term electric load forecasting", *Journal of Intelligent & Fuzzy Systems* vol.7, no.3, p.209-21, 1999.
10. Gorisek, J., "Distribution network automation utility experience", *10th International Conference on Power System Automation and Control*. PSAC'97, p.103-8, 1997.
11. Herrell, D., Beker, B., "Modeling of power distribution systems in PCs", *IEEE 7th Topical Meeting on Electrical Performance of Electronic Packaging*, p.159-62, 1998.
12. Hua Li, Hsiao-Dong Chiang, Gale, W.G., Bennett, J.T.H., "An incident based connectivity trace system for distribution network: algorithm and implementation", 1999, *IEEE Power Engineering Society Summer Meeting*.
13. Lestan, D., Gorisek, J., "Distribution network automation utility experience", *10th International Conference on Power System Automation and Control*, PSAC'97, p.103-8, 1997.
14. Huanga Ming-Yang, Chao-Shen Chena and Chia-Hung Linb, "Innovative service restoration of distribution systems by considering short-term load forecasting of service zones", *International Journal of Electrical Power &*

Energy Systems, Volume 27, Issues 5-6, June-July 2005, Pages 417-427

15. Mitchell Scott, An Extensive Examination of Data Structures Using C# 2.0, 4GuysFromRolla.com, Update,January2005, http://msdn.microsoft.com/vcsharp/default.aspx?pull=/library/enus/dv_vstechart/html/datastructures_guide.asp
16. Tae-I Choi, In-Seok Koh, Bok-Hyun Kim, Won-Soon Song, Wan-Sung Jang, "PC-based remote control and fault location system for distribution lin", IEEE Power Engineering Society, 1999 Winter Meeting, 1999.
17. Wainwright, I., "Engineering the benefits of a geographical information system: the business case for GIS". Wainwright, I. IEE Colloquium on Engineering the Benefits of Geographical Information Systems (Digest No.1997/105). IEE, London, UK. pp. 2/1-7, 1997.
18. Xing Weiguo; Yu Yixin, Xiao Jun, Wang Ruying, "A PC-based distribution AM/FM system", *ICEE '96. Proceedings of the International Conference on Electrical Engineering*, p.165-9 vol.1, 1996.
19. Yeh, E.-C., Tram, H., "Information integration in computerized distribution system planning", *IEEE Transactions on Power Systems*. Vol. 12, Iss. 2, pp. 1008-13, May 1997.
20. Al-A'ali, Mansoor, Improving the speed of electrical distribution network systems by improving data representation techniques *WSEAS Transactions on Circuits and Systems*, v 5, n 7, July, 2006, p 1124-1131.
21. Colas, Frederic Dieulot, Jean-Yves; Barre, Pierre-Jean; Borne, Pierre, Dynamics modeling and causal ordering graph representation of a non minimum phase flexible arm fixed on a cart, *WSEAS Transactions on Systems*, v 5, n 1, January, 2006, p 225-232.
22. Remy, Ghislain; Gomand, Julien; Barre, Pierre-Jean; Hautier, Jean-Paul, New current control loop with resonant controllers by using the causal ordering graph - Application to machine tools, *WSEAS Transactions on Systems*, v 5, n 1.
23. Reshmy, Krishnan R.; Srivatsa, S.K.; Ramakrishna, Implementation of frequent sub graph discovery *WSEAS Transactions on Information Science and Applications*, v 3, n 3, March, 2006, p 651-655.