論 文
50A-2-3

다익스트라 알고리즘을 이용한 배전계통의 향상된 사고복구 기법

Improved Service Restoration Technique by Using Dijkstra Algorithm in Distribution Systems

金 洛 京* · 金 載 哲** · 田 永 在* · 金 勳***
(Nark-Kyung Kim · Jae-Chul Kim · Young-Jae Jeon · Hoon Kim)

Abstract - This paper presents a fast and effective methodology for service restoration in large-scale distribution systems. The service restoration problem is formulated as a constrained optimization problem and requires the fast computation time and superior solution because the more unfaulted out-of-service area should be restored as soon as possible. The proposed methodology is designed to consider the fast computation time and priority service restoration by dijkstra algorithm and fuzzy theory in large-scale distribution systems. Simulation results demonstrate the validity and effectiveness of the proposed methodology on a 26-bus and 140-bus system.

Key Words : Service Restoration, Distribution Systems, Dijkstra Algorithm, Fuzzy Theory, Priority Restoration Area

1. INTRODUCTION

Customer satisfaction and service reliability are primary concerns in the power industry. Serval studies on utilities' experience suggest that customer satisfaction is closely correlated with service interruption frequency and duration. Service interruption duration can be significantly decreased via effective service restoration procedures.

The main objective in service restoration procedures is to restore as mush load as possible by transferring de-energized loads in the unfaulted out-of-service areas via network reconfiguration to other supporting feeders without violating operation and electrical constraints.

In the past, there have been considerable effort addressing service restoration. Castro et al. [1] suggested algorithms based on tree searching techniques utilizing switch tables that could be defined by an operator in 1980. Castro Jr. et al. [2] proposed a method in which they combined the service restoration and load balance between each feeder in 1985. Aoki et al. [3] presented an algorithm for load transfer by automatic sectionalizing switch operations in distribution systems on an outage occurrence subject to the transformer and line-capacity constraint in 1987. Also Aoki et al. [4] developed a method in which loads in an outage area were transferred to the transformers adjacent to the affected area based on current and voltage constraints in 1988. Aoki et al. [5] suggested a non-combinatorial algorithm based on the effective gradient method, also Aoki et al. [6] presented an another algorithm for emergency service restoration in which load restoration priority could be considered in 1989. N.D.R. Sarma et al. [7] presented a new technique, wherein the network was reduced by merging certain set of nodes together and this reduced network was analyzed for finding alternate paths of power supply to the affected load points in 1991. Y. Imamura et al. [8] suggested an application method of fuzzy evaluation for efficient line transfer in 1993. K. Song et al. [9] improved service restoration procedures of [8] in 1995. Y. H. et al. [10,11] and Q. Zhou et al. [12] presented the combined fuzzy theory and heuristic method for fast service restoration. J. Kim et al. [13] proposed an outage restoration method by Object-Oriented Programming in 2000.

The previous approaches have service restoration procedures without a numerical computation or are suitable for a small distribution system. From a computational viewpoint, the service restoration problem is a NP-complete problem. NP-complete problems typically require time consuming combinatorial optimization algorithms to solve for a global optimal solution. However, the global optimal solution is unnecessary for the service restoration because there may exist many solutions restoring power to out-of-service areas. For this reason, the service restoration procedure must be designed to consider the fast computation time

---

* 正 會 員 : 崇實大學校 電氣工學科 博士科程
** 正 會 員 : 崇實大學校 電氣工學科 敎授 · 工博
*** 正 會 員 : 崇實大學校 電氣工學科 碩士
接受日字 : 2000年 8月 25日
最終完了 : 2001年 2月 1日

and appropriate convergence property.

In this paper, we apply dijkstra algorithm, having the computation time less then other combinatorial optimization algorithms, to search an appropriate solution for service restoration in large-scale distribution systems. To satisfy a radial structure constraint, kruscal algorithm is included in the service restoration procedure. We also adopt fuzzy theory to restore priority restoration area when a distribution system is not restored perfectly. Numerical examples demonstrate the validity and effectiveness of the proposed methodology on a 26-bus and 140-bus system.

2. PROBLEM FORMULATION

The service restoration problem is formulated as a constrained multi-objective optimization problem. Because distribution systems are normally operated in radial structure, we can regard service restoration problem as a finding problem of minimum spanning tree in graph theory.

2.1 Define of service restoration terms

In the follows we define the number of terms commonly used in this paper in order to make the presentation of this paper clearer.

Tie switch : A normally open switch connecting two adjacent feeders.

Faulted area : A portion of the feeder which is experiencing a permanent fault.

Unfaulted out-of-service area : A portion of the feeder downstream from the faulted area which can be separated from the faulted area by opening a normally closed switch.

Adjacent feeder : A feeder with tie switches to the faulted or overloaded feeder.

Secondary feeder : A feeder with tie switches to the adjacent feeders but no tie switch to the faulted or overloaded feeder.

Boundary switch : A normally open switch that connects the unfaulted out-of-service area or the area with overloaded line sections.

2.2 Object Function and Constraints

In this paper, the objective function for the service restoration problem is can be written as follows.

$$minimize\ E = \sum_{(i,j)\in U} d_{ij} x_{ij} \quad (i < j) \tag{1}$$

where,

E : objective function
i, j : vertex or bus
U : set of all vertex or bus ($U=1,2,\cdots,n$)
d_{ij} : cost between i and j
x_{ij} : the decision variable between i and j (x_{ij}=0 or 1)

Equation (1), the objective function, consider constraints for a real solution of the service restoration problem. The constraints are radial structure, line capacity, and feeder capacity in this paper, which are represented in (2).

[Constraints]
(Flow constraint between i and k [i-j-k section])

$$\sum_{(k,i)\in U} x_{ik} - \sum_{(i,j)\in U} x_{ij} = \begin{cases} 1 : j \text{ is start point} \\ -1 : j \text{ is end point} \\ 0 : \text{otherwise} \end{cases} \tag{2.A}$$

(Line and Feeder capacity constraints)

$$\sum L_j \le b_i \ (i \in U,\ j \in D) \tag{2.B}$$

where,

i, j : vertex or bus
U : set of all vertex or bus ($U=1,2,\cdots,n$)
x_{ij} : the decision variable between i and j (x_{ij}=0 or 1)
D : set of all downstream bus from bus i
L_j : load amount at bus j
b_i : limit capacity at bus i

In above equation, the meaning of (2.A) is equal to radial operation condition in distribution systems, which is considered by kruskal algorithm that is one of the greedy search algorithm. The minimized objective function becomes the solution for service restoration while operation constraints is satisfied.

3. OVERVIEW OF DIJKSTRA AND KRUSCAL ALGORITHM

3.1 Dijkstra Algorithm

A graph $G=(V,E)$ is a collection of vertices V and edges E. A vertex is a point in space, and an edge is a connection between two vertices. We will also refer to a vertex as a node and, in the context of trellises, as a state. We will refer to a branch as a single edge. The edge between the vertices a and b is denoted as (a, b). A path from a vertex a to a vertex b is a list in which successive vertices are connected by edges in the graph; it is a list of adjacent branches. A weighted graph is one in which numerical weights are assigned to each edge. The cost of a path is the sum of the weights along the path.

Dijkstra algorithm is so useful method to solve the problem of finding the shortest path from a vertex in a graph to a destination. It turns out that one can find the shortest paths from a given vertex to all vertices in the graph, hence this problem is sometimes called the single-source shortest path problem.

Dijkstra algorithm works on the undirected and non-negative weight graph $G=(V,E)$. It finds shortest paths from the source vertex to all other vertices. The main idea of Dijkstra algorithm is to change temporary labels associated with vertices into permanent ones. The permanent label of a vertex denotes the shortest path weight from the source vertex to the current vertex. For vertex i, we denote as follows.

$$A(i)=\{j; e=(i,j)\in E, j \text{ has temporary label}\} \quad (3)$$

We allocate a permanent label 0 to a vertex s, a temporary label $C(s, j)$ to $j\in A(s)$, and a temporary label ∞ to all other vertices . Denote P to be the set containing all the vertices with permanent labels, and $T=V-P$ to be the set containing all the vertices with temporary labels. Steps of the dijkstra algorithm are shown as follows.

[STEP 1] Initialize a temporary label.

If starting vertex s has a connecting path to a vertex, temporary label become $C(s,j)$, the others become ∞.

[STEP 2] Select a shortest path.

Prepare a temporary label $C(s,j)$ and select a minimum value then selected temporary label set a permanent label.

[STEP 3] Compare a detour path.

Adjacent vertices of the determined permanent label reconstruct the value of the temporary label. Compared a temporary label to detour path.

[STEP 4] Ending.

If all vertexes are compared then Dijkstra procedure is ending, otherwise go to STEP 2

3.2 Kruskal Algorithm

Kruskal algorithm is that of finding a minimum spanning tree in a directed or undirected graph. This method is one of the greedy search algorithms, creates a forest of trees. Initially the forest consists of single vertex trees. At each step, edges are added to join two trees together. If it were to form a loop, it would simply link two vertices that were already part of a single connected tree. The basic algorithm looks like the following.

[STEP 1]

The forest is constructed with each vertex in a separate tree.

[STEP 2]

If it forms a cycle, reject it.

[STEP 3]

Else add it to the forest.

Adding it to the forest will join two trees together.

3.3 Dijkstra-Kruskal Algorithm

Dijkstra algorithm is suitable for a shortest path problem to find from the source vertex to all other vertices, but it has sometimes loops in solutions. Kruskal algorithm can surely find a spanning tree, is a radial structure condition in a distribution system, in networks. But the solution quality of kruskal algorithm is less then it of dijkstra algorithm. Therefore, we find a solution by dijkstra algorithm and then kruskal algorithm is performed on the solution. If the solution has not loops, the same solution can be obtained by kruskal algorithm. Otherwise, edges with higher weights in the solution is removed in kruskal algorithm procedure. For the example related to dijkstra and kruskal algorithm, a graph with six vertices and ten edges is shown in Fig. 1.

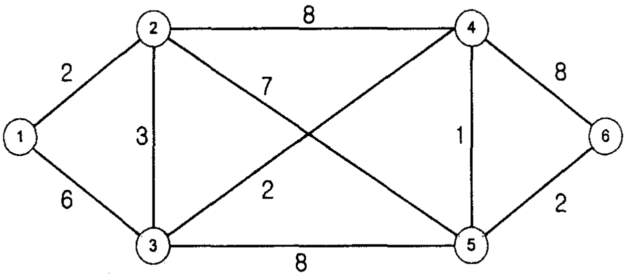


Fig. 1. The undirected graph with associated edge weights

For the representation of a graph, a adjacency matrix is usually used. If $G=(V,E)$ has m vertices labeled 1, 2, ... , m, then the adjacency matrix of the graph is the $m\times m$ matrix in which the i, jth element is the number of edges joining vertex i to vertex j. If the graph is undirected, then the adjacency matrix is symmetric. For the graph in Fig. 1, the adjacency matrix is follows.

Table 1. The adjacency matrix with Fig. 1

	1	2	3	4	5	6
1	0	2	6	∞	∞	∞
2	2	0	3	8	7	∞
3	6	3	0	2	8	∞
4	∞	8	2	0	1	8
5	∞	7	8	1	0	2
6	∞	∞	∞	8	2	0

At each step, the algorithm determine the vertex $i \in T$ with the minimum temporary label, the vertex $i \in T$ with the permanent label, the predecessor index of vertex i, and the temporary values of all the vertex $j \in A(s)$. This procedure is repeated until all vertices become permanent ones. In the end, the last result of this procedure has a shortest path in given weight graph $G=(V,E)$. The shortest path for Fig. 1 can be found by the above mentioned procedure as follows.

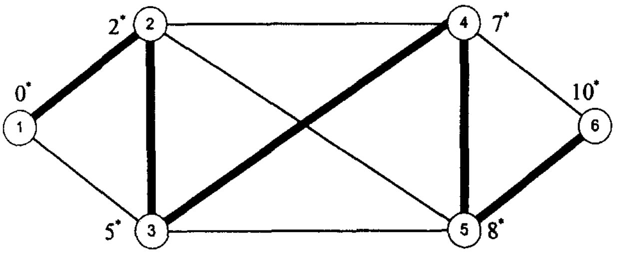


Fig. 2. The short path for Fig. 1

After a solution is obtained by dijkstra algorithm, kruskal algorithm is performed to check whether the solution has loops or not. For the example, we assume that the graph in Fig. 3 is obtained by dijkstra algorithm.

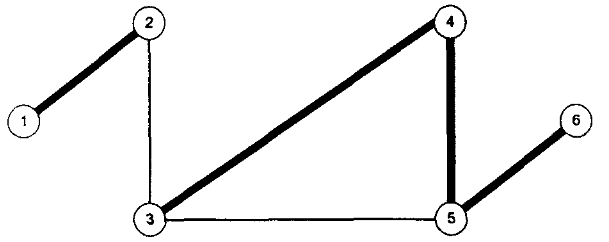


Fig. 3. The graph with a loop

As shown in Fig. 3, this graph has a loop that is configured by three edges(3-4, 3-5, and 4-5). In Fig. 3, thick lines are obtained by kruskal algorithm procedure, and one edge must be added to make a spanning tree. In the procedure of kruskal algorithm, the selected all vertices are included in trees. It is assume that vertex 1 and 2 are included in tree 1 and vertex 3, 4, and 5 are included in tree 2, respectively. If the edge between vertex 3 and 5 in the same tree is added, a loop is created and then this selection is rejected. Therefore, a spanning tree can be certainly obtained by kruskal algorithm after the performance of dijkstra algorithm.

4. PROPOSED ALGORITHM

4.1 The weigh decision for dijkstra algorithm

We must determine the weight of edges to find shortest paths by dijkstra algorithm in distribution system, which requires loads of bus i and j to change non-negative number between bus i and j. Much of the available data relating to loads of a bus are neither deterministic nor probabilistic. For instance, statements such as "load at bus A is approximately 2MW" or "load at bus B is mainly industrial type" are clearly neither deterministic nor probabilistic. To handle linguistic variables of knowledge representation used by domain experts, the fuzzy set approach provides a means to deal with this kind of imprecision or uncertainty. In this paper, each load type is divided into five load levels very small(VS), small(S), medium(M), large(L), and very large(VL), which are associated the membership function $\mu(x)$ in (4).

$$\mu(x) = \frac{1}{1+(\frac{x-m}{\alpha})^2}, \quad x \gg 0 \tag{4}$$

Each load level has its own parameters m and α for the membership function $\mu(x)$. Note that m, which is usually denoted as the mean of the function, gives the most possible value of the fuzzy variable. The other parameter α, which is called the spread of the function, describes the degree of fuzziness with the fuzzy variable. In this paper, the parameters m and α for the membership functions of the five linguistic variables VS, S, M, L and VL are based on parameters in [11], which are modified for the proposed methodology and summarized in Table 2.

Table 2. Summary of membership function parameters m and α

parameters		linguistic variables				
		VS	S	M	L	VL
m	commercial	0.130	0.280	0.570	0.740	0.850
	industrial	0.150	0.350	0.540	0.720	0.900
	residential	0.120	0.220	0.450	0.680	0.860
α	commercial	0.022	0.056	0.057	0.074	0.176
	industrial	0.017	0.033	0.048	0.069	0.135
	residential	0.025	0.062	0.092	0.096	0.243

The weight between bus i and j, the membership function of the sum of two variables, is obtained by (5).

$$\mu_{ij}(k) = \frac{1}{1+\left(\frac{k-m_{ij}}{\alpha_{ij}}\right)^2} \qquad (5)$$

where

$k = i + j$

$m_{ij} = m_i + m_j$

$\alpha_{ij} = \alpha_i + \alpha_j$.

In this paper, the weight between bus i and j is determined by loads and load types of the bus, which can restore priority restoration area by using fuzzy parameters when distribution system is not restored perfectly. Fuzzy parameters depend on the exact load estimation and operator's experience, which may significantly affect the system topology in large-scale distribution systems. Therefore, the service restoration procedure may be obtained different solutions as a daily load curve and the information of priority restoration areas in the same case.

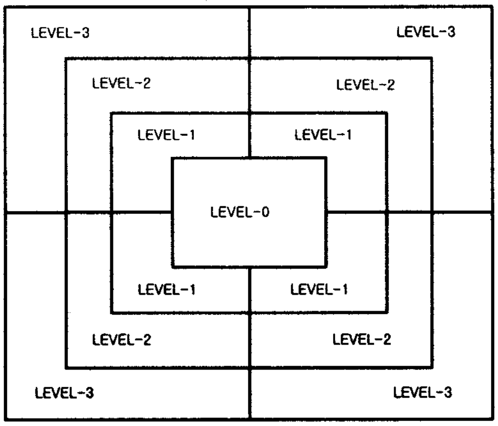
4.2 Procedure of the proposed methodology

In this paper, we handle a mathematical method to solve the formulated problem that is to decide the least cost path for service restoration. In order to solve the service restoration problem effectively, we propose the methodology based on dijkstra algorithm that is most effective to solve the shortest path problem. To perform the dijkstra algorithm, weights on the graph are determined by loads and load types. Dijkstra algorithm searches a optimal solution for service restoration by using the weight and adjacency matrix. But dijkstra algorithm sometimes finds the graph with loops, so kruskal algorithm is performed to find the complete solution from the graph with loops.

Outage such as multiple faults and transformer faults, which have large effect, requires many switching numbers, while outage in the feeder with light loads and the end of a feeder is relieved by simple operation. For this reason, we use "level" concept to reduce the computation time.

As shown in Fig. 4, level-0 is the faulted area. At first, the proposed methodology considers a level-1 feeder and then search an optimal solution. If unfaulted out-of servoce area is not restored, level-1 is step up to level-2. The procedure is repeatedly proceeded until level-3 as previous procedure. Faulted area is restored by the load transfer operation through from level-0 to level-3.

The above mentioned procedure consists of the simple comparison and accumulation operation and the flowchart of the proposed methodology is shown in Fig. 5.



LEVEL-0 : Faulted Area
LEVEL-1 : Adjacent Feeder
LEVEL-2 : Adjacent Feeder with LEVEL-1 Feeder
LEVEL-3 : Adjacent Feeder with LEVEL-2 Feeder

Fig. 4. Level concept for service restoration

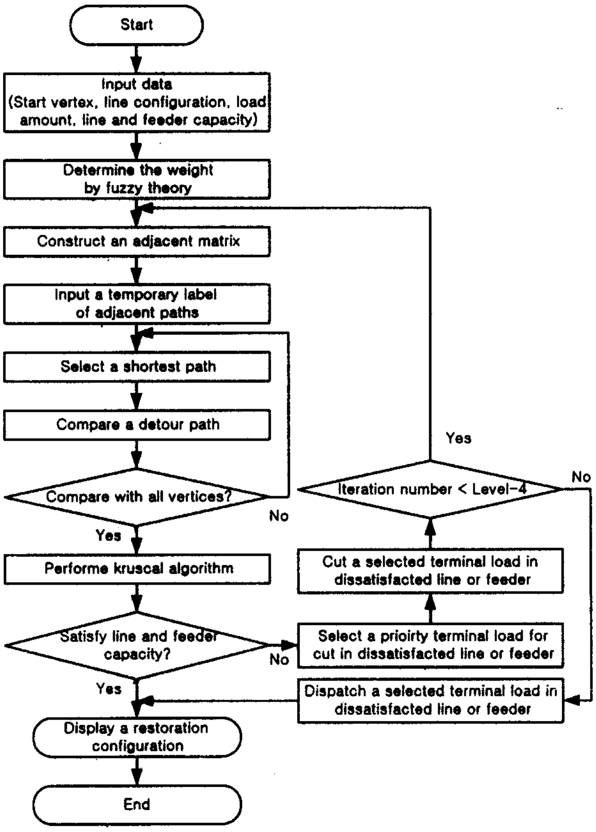


Fig. 5. Flowchart of the proposed methodology

5. NUMERICAL SIMULATION

The proposed methodology is implemented in *C* language on an *Intel Pentium II* 350*MHz Processor* with an adjacency matrix to represent the database of the distribution systems. The proposed methodology is simulated for the effectiveness and validity on a 26-bus and 140-bus system.

5.1 26-bus system

This system consists of 3 feeders, 26 buses and 31 branches as shown in Fig. 6.

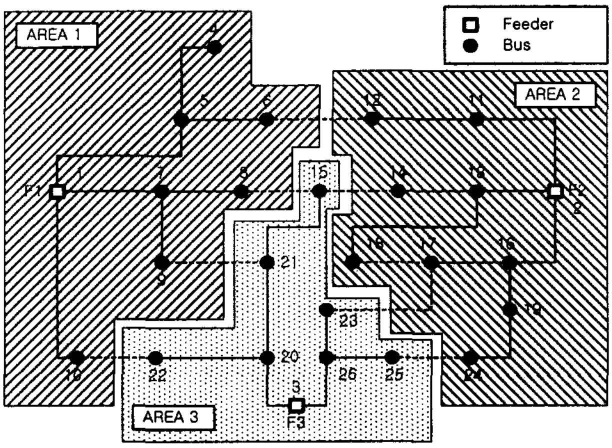


Fig. 6. Initial configuration of a 26-bus system

The proposed methodology simulate some general cases with the same fuzzy parameter which is regardless of load types, which are primary and secondary adjacent outages, and priority restoration cases. For simulation results, all secondary adjacent outages are restored in our simulation, but the four primary adjacent outages cannot be restored for feeder and line capacity constraints. The simulation results are compared with results of previous approaches [8,9,14], which are shown in Table 3.

Table 3. Comparison result with other approaches

Outage Section	Reference [8]		Reference [9]		Reference [14]		Proposed Methodology	
	SF	SFA	SF	SFA	SF	SFA	SF	SFA
1-5	4	2	4	2	4 5	2 4	4	2
1-7	7 9	5 8	7	5	8	10	7	5
1-10	-	-	-	-	-	-	-	-
2-11	11	4	-	-	-	-	-	-
2-13	13	2	-	-	-	-	-	-
2-16	-	-	-	-	-	-	-	-
3-20	20	3	-	-	20	3	-	-
3-26	25 26	6 4	26	4	25 26	6 4	26	4
4-5	4	2	4	2	4	2	4	2
20-21	-	-	-	-	-	-	-	-

SF : Service Failure Area

SFA : Service Failure Amount [MW]

Service restoration procedures in [8] and [14] cannot consider all alternative solutions, which is reason that some cases cannot be restored. Four cases cannot be certainly restored for feeder and line constraints and the other all cases can be completely restored through the proposed methodology, which show that the proposed methodology is suitable for service restoration in distribution systems.

We also handle a priority restoration case in a 26-bus system. To show the simulation result for priority service restoration, we assume the outage between bus 1 and bus 7 as shown in Fig. 7.

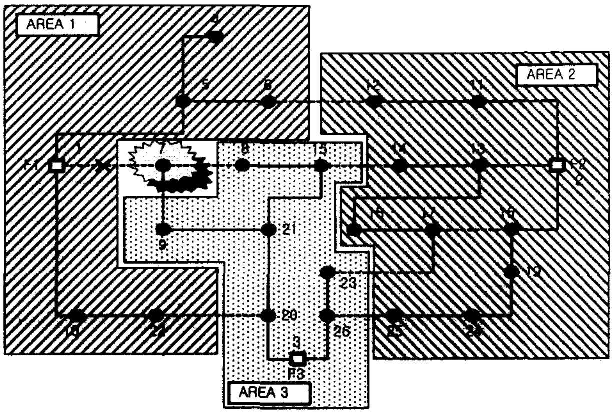


Fig. 7. The result of the outage between bus 1 and 7

In Fig. 7, bus 7 cannot be restored. If bus 7 is a priority restoration area, bus 7 should be preferentially restored. Fuzzy parameters are appropriately determined for the priority restoration procedure in hard constraints of this system. The proposed methodology finds a system topology as shown in Fig. 8.

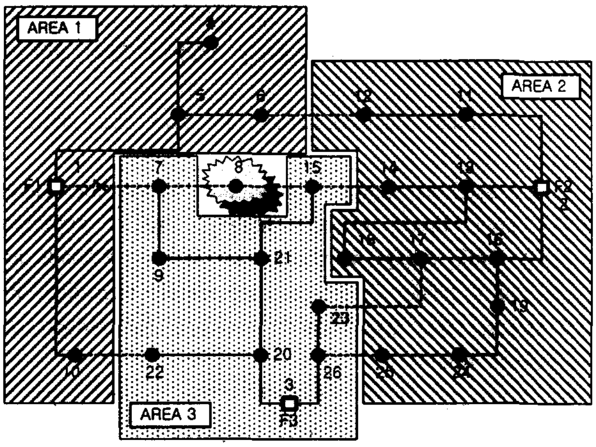


Fig. 8. The result of priority service restoration

5.2 140-bus system

This distribution system is a real system of *K* city in kyong-gi province. The real distribution system model consist of 2 transformers, 8 feeders, and 140 buses. Loading ratio is set 80% of full loads for various simulation results from this simulation. Also, this system is composed of CNCV 325㎟, ALOC 160㎟ and ALOC 95 ㎟. Initial topology of the 140-bus system is shown in Fig. 9.

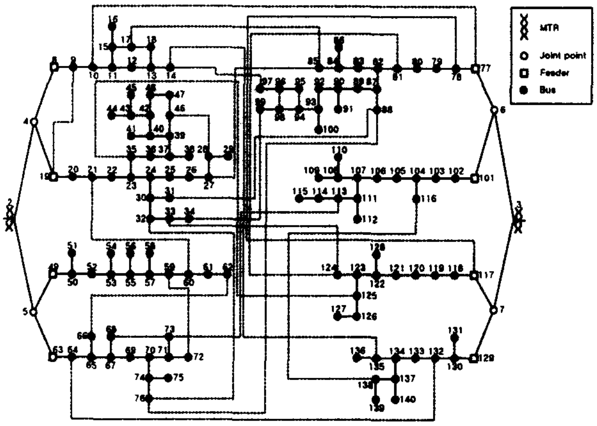


Fig. 9. The initial configuration of the 140-bus system

We simulate the outage between bus 20 and bus 21 and the result of this case is shown in Fig. 10. As shown in Fig. 10, unfaulted out-of-service area is divided into three small groups and restored by boundary switches with adjacent feeders. The system is satisfied with constraints of distribution system which are line, feeder capacity and radial operation condition. The computation time takes 0.301 seconds for the result of this case.

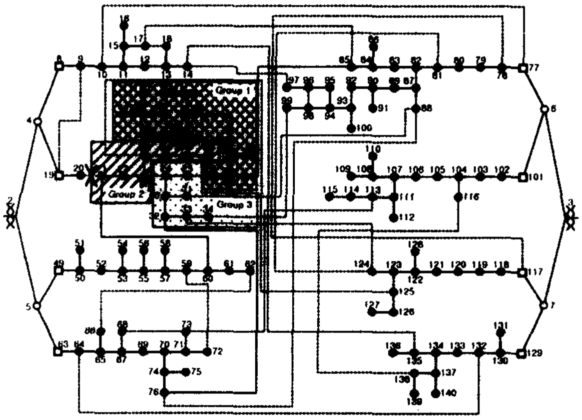


Fig. 10. The configuration for the outage between bus 20 and bus 21.

Also, we simulate the case of priority service restoration. In this case, we allocate 90% of full load to this system for the occurrence of restoration failure areas. This case is the same with the outage between bus 20 and bus 21 as shown in Fig. 10. The restoration failure result of this case is shown in Fig. 11. When outage is not restored completely, the proposed methodology preferentially restore the bus with commercial and industrial types. For this reason, various load types are considered by fuzzy parameters in Table 2, which can restore priority restorative area. As shown in Fig. 11, bus 41 is not restored when all bus is assumed residential type. After all bus are assigned to various load type, bus 41 with industrial type is restored and bus 44 with residential type is not restored by the proposed methodology as shown in Fig. 12. The computation time of this case takes 0.451 seconds.

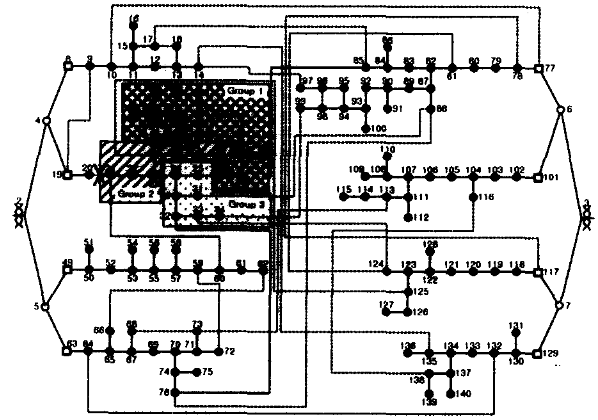


Fig. 11. Failure configuration for the outage between bus 20 and 21 when all bus is residential type

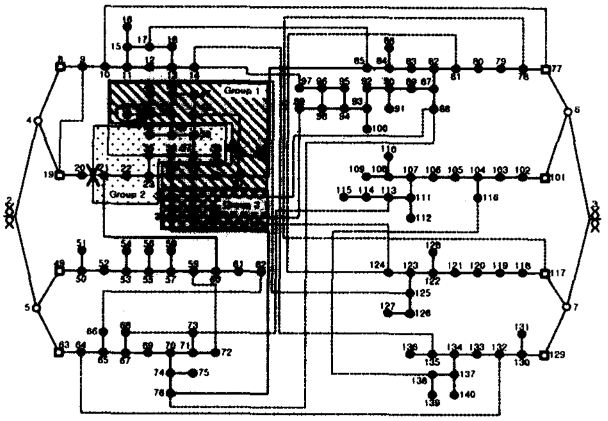


Fig. 12. Priority restoration result for the outage between bus 20 and 21 by the proposed methodology.

The proposed methodology is successfully applied on a 26-bus and 140-bus system and has the computation time under 1 second. Also priority restoration can be performed by adjusting fuzzy parameters with load compositions and types.

6. CONCLUSION

In this paper we propose an efficient methodology for service restoration in large-scale distribution systems. The main object in service restoration is to restore as mush loads as possible in the acceptable computation time when an outage is occurred in distribution systems. Dijkstra algorithm searches a good solution for service restoration in distribution systems and kruscal algorithm is used for satisfying the radial structure constraint. Also, fuzzy theory is used for determining the weight of edges in dijkstra algorithm, which can preferentially restore priority restoration areas when distribution system is not restored perfectly. Simulation results demonstrate the validity and effectiveness of the proposed methodology on a 26-bus and 140-bus system.

감사의 글

본 연구는 기초전력공학공동연구소의 "AI 기법을 이용한 배전 계통의 최적 라우팅 기법 개발"에 따른 연구비(97-중-05)로 수행되었음.

REFERENCE

[1] C. H. Castro, J. B. Bunch, and T. M. Topka, "Generalized Algorithms for distribution Feeder Deployment and Sectionalizing," *IEEE Trans. on PAS*, Vol. PAS-99, No. 2, pp. 549-557, March/April 1980.

[2] C. A. Castro Jr. and A. L. M. Franca, "Automatic Power Distribution Reconfiguration Algorithm Including Operating Constraints," *Proceeding IFAC Conference on Electric Energy Systems, Brazil*, pp. 155-160, 1985.

[3] K. Aoki, H. Kuwabara, T. Satoh, and M. Kanezashi, "Outage State Optimal Load Allocation by Automatic Sectionalizing Switches Operation in Distribution System," *IEEE Trans. on Power Delivery*, Vol. 2, No. 4, pp. 1177-1185, October 1987.

[4] K. Aoki, T. Satoh, M. Itoh, H. Kuwabara, and M. Kanezashi, "Voltage Drop Constrained Restoration of Supply by Switch Operation in Distribution Systems," *IEEE Trans. on Power Delivery*, Vol. 3, No. 3, pp. 1267-1274, July 1988.

[5] K. Aoki, K. Nara, M. Itoh, T. Satoh, and H. Kuwabara, "A New Algorithm for Service Restoration in Distribution Systems," *IEEE Trans. on Power Delivery*, Vol. 4, No. 3, pp. 1831-1839, July 1989.

[6] K. Aoki, K. Nara, and T. Satoh, "New Configuration Algorithm for Distribution System-Priority Constrained Emergency Service Restoration," *Proceeding of IFAC Conference on Power Systems and Plant Control, Seoul, Korea*, pp. 443-448, 1989.

[7] N. D. R. Sarma, V. C. Prasad, K. S. Prakasa Rao, and V. Sankar, "A New Network Reconfiguration Technique for Service Restoration in distribution Networks", *IEEE Summer Meeting, San Deigo*, USA, July 1991.

[8] Y. Imamura, M. Kanoi, and J. Kawakami, "Fuzzy Inference Application to Planning of Distribution Network Switching," *The Transactions of The Institute of Electrical Engineers of Japan*, Vol. 113-B, No. 5, pp. 493-498, 1993.

[9] K. Song, Y. Kim, B. Lee, J. Cha, J. Namkung, S. Lee, "Emergeny Service Restoration Algorithm Using Efficient Reconfiguration in Distribution System," *The Transactions of The Korean Institute of Electrical Engineers*, Vol. 44, No. 7, pp. 843-851, July 1995.

[10] Y. Hsu and H. Kuo, "A Heuristic Based Fuzzy Reasoning Approach for Distribution Systems Service Restoration," *IEEE Trans. on Power Delivery*, Vol. 9., No. 2, pp. 948-953, April 1994.

[11] H. Kuo and Y. Hsu, "Distribution System Load Estimation and Service Restoration Using a Fuzzy Set Approach," *IEEE Trans. on Power*, Vol. 8, No. 4, pp. 1950-1957, October 1993.

[12] Q. Zhou, D. Shirmohammadi, and W. E. Liu, "Distribution Feeder Reconfiguration For Service Restoration and Load Balancing," *IEEE Trans. Power System*, Vol. 12, No. 2, pp. 724-729 May 1997.

[13] N. D. R. Sarma, Soumen Ghosh, and K. S. Prakass Rao, "Real Time Service Restoration in Distribution Networks a Practical Approach," *IEEE Trans. on Power Delivery*, Vol. 9, No. 4, pp. 2064-2070, October 1994.

[14] J. Kim and Y. Baek, "Service Restorative System in Emergency State using Object-Oriented Structure," *The Transactions of The Korean Institute of Electrical Engineers*, Vol. 49, No. 1, pp. 1-5, January 2000.

[15] J. Dombi, "A general class of fuzzy operators, the De Morgan class of fuzzy operators and fuzziness measures induced by fuzzy operators", Fuzzy Sets and Systems, Vol. 11, pp.115-134, 1983.

[16] M. A. Weiss, *Data Structures and Algorithm Analysis in C*, The Benjamin/CummingsPublishing Company, California, USA, 1993.

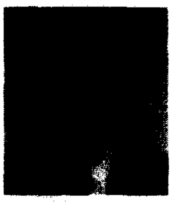
저 자 소 개

김 낙 경 (金 洛 京)
1953년 6월 24일 생. 1984년 인천대 전기공학과 졸업. 1994년 숭실대 대학원 전기공학과 졸업(석사). 1998~현재 동 대학원 전기공학과 박사과정. 주식회사 대동기술단 대표이사.

Tel : 02-332-3329
E-mail : nkkim@dd-eng.com

김 재 철 (金 載 哲)
1955년 7월 22일생. 1979년 숭실대 전기공학과 졸업. 1983년 서울대 대학원 전기공학과 졸업(석사). 1987년 서울대 대학원 전기공학과 졸업(공박). 1988년~현재 숭실대 공대 전기공학과 교수.

Tel : 02-820-0647
E-mail : jckim@ee.ssu.ac.kr



전 영 재 (田 永 在)
1972년 6월 30일 생. 1995년 숭실대 전기공학과 졸업. 1997년 숭실대 대학원 전기공학과 졸업(석사). 현재 동 대학원 전기공학과 박사과정

Tel : 02-817-7966
E-mail : anneal@dreamwiz.com

김 훈 (金 勳)
1971년 9월 7일 생. 1995 서울산업대 전기 공학과 졸업. 2000년 숭실대 대학원 전기 공학과 졸업(석사). 현재 삼성전자 생활가전총괄 리빙사업부

Tel : 031-200-6836
E-mail : hoon_kim@samsung.com