# PC Cluster based Parallel Adaptive Evolutionary Algorithm for Service Restoration of Distribution Systems

**Kyeong-Jun Mun*, Hwa-Seok Lee†, June Ho Park**, Hyung-Su Kim*** and Gi-Hyun Hwang§**

**Abstract** - This paper presents an application of the parallel Adaptive Evolutionary Algorithm (AEA) to search an optimal solution of the service restoration in electric power distribution systems, which is a discrete optimization problem. The main objective of service restoration is, when a fault or overload occurs, to restore as much load as possible by transferring the de-energized load in the out of service area via network reconfiguration to the appropriate adjacent feeders at minimum operational cost without violating operating constraints. This problem has many constraints and it is very difficult to find the optimal solution because of its numerous local minima. In this investigation, a parallel AEA was developed for the service restoration of the distribution systems. In parallel AEA, a genetic algorithm (GA) and an evolution strategy (ES) in an adaptive manner are used in order to combine the merits of two different evolutionary algorithms: the global search capability of the GA and the local search capability of the ES. In the reproduction procedure, proportions of the population by GA and ES are adaptively modulated according to the fitness. After AEA operations, the best solutions of AEA processors are transferred to the neighboring processors. For parallel computing, a PC cluster system consisting of 8 PCs was developed. Each PC employs the 2 GHz Pentium IV CPU and is connected with others through switch based fast Ethernet. To show the validity of the proposed method, the developed algorithm has been tested with a practical distribution system in Korea. From the simulation results, the proposed method found the optimal service restoration strategy. The obtained results were the same as that of the explicit exhaustive search method. Also, it is found that the proposed algorithm is efficient and robust for service restoration of distribution systems in terms of solution quality, speedup, efficiency, and computation time.

**Keywords**: Adaptive Evolutionary Algorithm (AEA), Distribution System Service Restoration, Evolution Strategy (ES), Genetic Algorithm (GA), PC cluster system

## 1. Introduction

In recent years, because of the quickly growing size and complexity of distribution systems, rapid restoration of out-of-service areas and a speedy return to normal operating conditions following a fault has become more critical in order to improve distribution system reliability. Also, in distribution system operation, service interruption frequency, interruption duration, power loss and voltage drop should be reduced in order to increase customer satisfaction by the introduction of automation techniques.

The main objective of service restoration is, when a fault or overload occurs, to restore as much load as possible by transferring the de-energized load in the out of service area via network reconfiguration to the appropriate adjacent feeders at minimum operational cost without violating operating constraints. The objective function is usually defined as the number of switching operations, the minimal amount of the unreserved load, the load balance of transformers and feeders, etc. The constraints that should be considered are: radial configuration, feeder loading, voltage drop, main transformer loading, and line capacity.

The service restoration problem is a complicated combinatorial optimization problem with non-linear constraints, because the number of variables and constraints involved in restoration planning is large. To resolve this problem, various approaches have been investigated. These algorithms are based on knowledge of the features of distribution systems. Heuristic approaches [1], a branch exchange method [2], an expert system [3], machine learning [4], AI technique [5], neural network [6] and fuzzy reasoning [7] have been developed to determine restoration plans quickly. However, these methods are likely to produce sub-optimal solutions.

† Corresponding Author: Dept. of Electrical Engineering, Koje College, Korea. (hslee@koje.ac.kr)
* Korea Atomic Energy Research Institute (KAERI), Korea. (kjmun@kaeri.re.kr)
** School of Electrical and Computer Engineering, Pusan National University, Korea. (parkjh@pusan.ac.kr)
*** Post-Doctor at Illinois Institute of Technology, USA. (kimhy@iit.edu)
§ Division of Internet Engineering, Dongseo University, Korea. (hwanggh@dongseo.ac.kr)
Received October 11, 2005 ; Accepted March 8, 2006

In parallel computing, problems are divided into several sub problems, and allocated to each processor. This greatly reduces computation time and enhances computation efficiency [8]. To realize the parallel algorithm, parallel computers such as the transputer were utilized but these computers are quite costly to use. Recently, PC clustering, one of the types of parallel or distributed processing systems that is composed of a collection of interconnected workstations or PCs working together as a single, integrated computing resource has been used for parallel computing [9].

Evolutionary algorithms (EA) are based on the principles of genetics and natural selection. Among EAs, GA simulates the crossover and mutation of natural systems, giving it a global search capability [10], whereas, evolution strategy (ES) simulates the evolution of an asexually reproducing organism. ES can find a global minimum, in case of combining another EA, and it could also be used as an efficient local search technique [11]. In the conventional method described above, parameter values and operator probabilities for the GA and ES are adapted to determine a solution and find it efficiently [12-15]. For GA, the population size, crossover rate, mutation rate and operation method are adaptively modified in each generation [12, 13]. To enhance the performances of the ES, mutation parameters are adapted during the run in ES [14, 15].

In this paper, we propose an Adaptive Evolutionary Algorithm (AEA), which is an algorithm in which the ratio of population to which the GA and ES will adjust is adaptively modified in the process of reproducing according to fitness. We use ES to optimize locally, while the GA optimizes globally. In other words, the resulting hybrid scheme produces improved reliability by exploiting the "global" nature of the GA as well as the "local" improvement capabilities of the ES. In this investigation, parallel AEA was developed for the service restoration of distribution systems. After AEA operations for each processor, the best solution of each AEA processor is transferred to the neighboring processors. For parallel computing, a PC cluster system consisting of 8 PCs was developed. Each PC employs a 2 GHz Pentium Ⅳ CPU, and is connected with others through switch based fast Ethernet.

The proposed algorithm can be used in the event of a fault or overload of transformers and feeders. The output of the algorithm is a N best list of candidate solutions and the corresponding switching sequence. An index of each result helps operators to make the final decision on how to restore the out-of-service area. The developed service restoration algorithm has been tested on a practical distribution system in Korea. The obtained results were compared to that of the explicit exhaustive search method [1]. It was found from the test results that the proposed method is capable of achieving a proper restoration plan in a very efficient manner.

## 2. Service Restoration of the Distribution System

The main objective of service restoration is, when a fault or overload occurs, to restore as much load as possible by transferring de-energized loads in the isolated areas to other feeders that can take on the extra load. When transferring loads, candidate solutions are evaluated based on 2 criteria: the number of switching operations, and load balancing to minimize the risk of overload. Also, one has to check whether the operating constraints are satisfied. Since the feeders in practice are usually operated radially, the radial of the system topology is an important constraint in the problem. There are also constraints that must be considered for each feeder: voltage drop limits and line/transformer capacity limits. The integral objective function and each element for the service restoration problem are defined by the following expression:

$$obj = p_1 \cdot IC^h + p_2 \cdot ILB^h + p_3 \cdot IP^h + p_4 \cdot IV^h \quad (1)$$

where, $IC^h$ : no. of switching operation

$ILB^h$ : load balancing index

$IP^h$ : power of unrestored load

$IV^h$ : voltage drop constraint index

$p_1, p_2, p_3, p_4$ : weighting factors

$h$ : h-th solution

a) No. of switching operation: Less switching operation makes it possible to restore the out-of-service area quickly and prevents shortening of durability.

$$IC^h = N_s \quad (2)$$

where, $N_s$ : no. of switching operation

b) Load balancing index: Load balancing of the tie feeder after the service restoration to minimize the risk of overload.

$$ILB^h = \begin{cases} \exp(-Max(y_{cap} - y_{\max})/\alpha_{ILB} & if \quad y_{\max} > 0{,}75\, y_{cap} \\ 0.1 \cdot \exp(-Max(y_{cap} - y_{\max})/\alpha_{ILB} & otherwise \end{cases} \quad (3)$$

where, $y_{cap}$ : the capacity of the tie feeder

$y_{\max}$ : maximum possible tie-feeder load after restoration

$\alpha_{ILB}$ : constant

c) Power of unrestored load: Power of unrestored load consists of the normal and priority customer load.

$$IP^h = p\sum_{i\in\alpha_p^h} P_i^h + q\sum_{i\in\alpha_q^h} p_i^h \quad (4)$$

where, $\alpha_p^h$, $\alpha_q^h$ : set of normal and priority customer

$p, q$ : weighting factor

$P_i^h$ : i-th section load

d) Voltage drop constraint index

$$IV^h = \underset{i}{Max}[\exp(\alpha_{IV}\cdot\frac{V_r^n - V_{ij}^h}{V_r^n}) - 1] \quad (5)$$

where, $V_r^n$ : feeder voltage

$V_{ij}^h$ : voltage of j-th section of i-th feeder,

$i = 1, 2, \cdots, n_f$ , $j = 1, 2, \cdots, n_{si}$

$n_f$ : no. of feeder

$n_{si}$ : no. of section of i-th feeder

$\alpha_{IV}$ : constant

The constraints for the service restoration problem are defined by the following expression:

a) Main transformer loading constraint

$$S_{TSi}^h \le S_{TSi}^n \quad (6)$$

where, $n_{TS}$ : no. of main transformers, $i = 1, 2, \cdots, n_{TS}$

$S^n{}_{TSi}$ : max capacity of i-th main transformer

$S^h{}_{TSi}$ : current capacity of i-th main transformer

b) Feeder loading constraint

$$I_{ij}^h \le I_{ij}^n \quad (7)$$

where, $I^h{}_{ij}$ : current of j-th section in i-th feeder

$I^n{}_{ij}$ : current ratings of j-th section in i-th feeder

## 3. Parallel Adaptive Evolutionary Algorithm Using PC Clustering

GA, one of the probabilistic optimization methods, is robust and it is able to solve complex and global optimization problems. But the disadvantage is that it can suffer from excessive computation time before providing an accurate solution because of minimally using prior knowledge and not exploiting local information [10]. ES, which simulates the evolution of an asexually reproducing organism, is efficient in its local search capabilities. However, to solve complex problems, it forms hybrid EA [15].

In this paper, to reach the global optimum accurately and reliably in a short execution time, we designed AEA by bringing together pieces of the GA and ES. In AEA, GA operators and ES operators are applied simultaneously to the individuals of the present generation to create the next generation. Individuals with higher fitness value have a higher probability of contributing one or more chromosomes to the next generation. This mechanism gives greater rewards to either the GA operation or the ES operation depending on what produces superior offspring. To enhance the global search capability, the best solutions of each AEA-based node are transferred to the corresponding AEA-based node at specified iteration. For parallel computing, a PC cluster system consisting of 8 PCs was developed in this investigation.

### 3.1 PC Cluster System

Since the mid 1980s, high performance computers have been needed according to the development of large-scale science and engineering. Since supercomputers are expensive, cluster systems were created to replace supercomputers because of their availability of inexpensive high performance PCs, high speed networks, and development of integrated circuits.

PC cluster systems provide higher availability as well as enhanced performance by lower cost through the interconnection of several PCs or workstations. PC cluster systems are very competitive with parallel machines in terms of the ratio of cost to performance because clustering is one of the types of parallel or distributed processing systems, which is composed of a collection of interconnected low cost PCs working together as single and integrated computing resources. Also, it is easy to add nodes that construct the PC cluster. A basic construction diagram for the PC cluster is shown in Fig. 1.

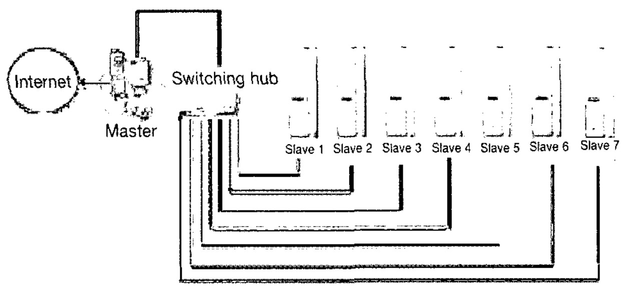



**Fig. 1.** Structure of PC cluster system

The performance of the PC cluster system depends on the quality of the message passing system, libraries, and compilers for parallel programming and performance of individual nodes. Therefore, it is important to select each component described above properly to obtain superior performance. The PC cluster system implemented in this paper is composed of 8 nodes based on fast Ethernet with an Ethernet switch. For the operating system, the master node uses a Windows 2000 server, and slave nodes use Windows 2000 pro. To connect each node, fast Ethernet cards and switching hubs were used. In data communication, an MPI library was utilized, which is effective for parallel application by using the message-passing method through TCP/IP over the Internet. Symantec PC-anywhere was used for remote control of each node, and MS visual C++ 6.0 was used for compilers of parallel programming. Table 1 describes the specifications of the 8-node PC cluster system developed in this paper.

**Table 1.** Specifications of 8-node PC cluster system

| | |
|---|---|
| CPU | Intel 2.0 GHz |
| Mother Board | LeoTech P4XFA |
| Chipset | VIA P4X266A |
| RAM | DDR SD RAM 256MB |
| HDD | Samsung 40GB 5600rpm |
| NIC | 3Com 3CSOHO 100-TX |
| Network Switch | 3Com 3C16465C Switch |
| Operating System | Window 2000 Server<br>Window 2000 Pro |
| MPI Library | MPICH 1.2.5 |
| Compiler | Visual C++ 6.0 |

### 3.2 Parallel Adaptive Evolutionary Algorithm

In AEA, the number of individuals created by the GA and ES operations is changed adaptively. An individual is represented as a real-valued chromosome, which makes it possible to hybridize GA and ES operations.

For AEA to self-adapt its use of GA and ES operators, each individual has an operator code to determine which operator to use. Suppose a '0' refers to GA, and a '1' to ES. At each generation, if it is more beneficial to use the GA, more '0's should appear at the end of individuals. If it is more beneficial to use the ES, more '1's should appear. After reproduction by roulette wheel selection according to the fitness, GA operations (crossover and mutation) are performed on the individuals that possess the operator code of '0' and the ES operation (mutation) is performed on the individuals that have an operator code of '1'. Elitism is also used. The best individual in the population reproduced both the GA population and ES population in the next generation. The major procedures of AEA are as follows:

**(1) Initialization**: The initial population is generated randomly. For each individual, an operator code is randomly initialized. According to the operator code, GA operations are performed on the individuals with operator code '0', while ES operation is applied where the operator code is '1'.

**(2) Evaluation and Reproduction**: Using the selection operator, individual chromosomes are selected in proportion to their fitness, which is evaluated using an objective function. After reproduction, GA operations (crossover and mutation) are performed on the individuals having an operator code of '0' and the ES operation (mutation) is performed on the individuals having an operator code of '1'. At every generation, the percentages of '1's and '0's in the operator code indicate the performance of the GA and ES operators.

**(3) Preservation of Minimum Number of Individuals**: At each generation, AEA may fall into a situation where the percentage of the offspring by one operation is nearly 100% and the offspring by other operation dies off. Therefore, it is necessary for AEA to preserve a certain amount of individuals for each EA operation. In this paper, we randomly change the operator code of the individuals with a higher percentage until the number of individuals for each EA operation becomes higher than a certain amount of individuals to be preserved. The predetermined minimum number of individuals to be preserved is set to 20% of the population size.

**(4) GA and ES**: The real-valued coding is used to represent a string of population. Modified simple crossover and uniform mutation are used as genetic operators. The modified simple crossover operator is defined as follows: if 2 strings, $S_v^t$ and $S_w^t$ are selected for the crossover operation and the crossover point is selected at the k-th component of the individual, the resulting offspring are defined as the combination of two vectors (individuals) as presented in Eq. (8)-Eq. (9).

<before crossover> <after crossover>

$$S_v^t = [v_1, \cdots, v_k, \cdots, v_N] \quad S_v^{t+1} = [v_1, \cdots, v_k', v_{k+1}', \cdots, v_N'] \quad (8)$$

$$S_w^t = [w_1, \cdots, w_k, \cdots, w_N] \quad S_w^{t+1} = [w_1, \cdots, w_k', w_{k+1}', \cdots, w_N'] \quad (9)$$

crossover point

where, $v_j' = \alpha_1 \times v_j + \alpha_2 \times w_j$

$w_j' = \alpha_1 \times w_j + \alpha_2 \times v_j$

$\alpha_1, \alpha_2$ : Random value between 0 and 1

$v_j, w_j$ : upper and lower bound of each variable

$N$ : no. of variables

In uniform mutation, we select a random gene k in an individual. If an individual and the k-th component of the individual is the selected gene, the resulting individual is as follows:

$$S_v^t = [v_1,\cdots, v_k,\cdots,v_N] \quad S_v^{t+1} = [v_1,\cdots,v'_k, v_{k+1},\cdots,v_N] \tag{10}$$

mutation point

where, $v'_k$ : Random value between upper bound and lower bound of $k$-th variable

Mutation is performed independently on each vector element by adding a normally distributed Gaussian random variable with mean zero and standard deviation (σ), as shown in Eq. (11). After adapting the mutation operator for ES population, if the improved individuals of the past generation are fewer than the present generation, standard deviation decreases in proportion to the decrease rates of standard deviation ($c_d$), otherwise, standard deviation increases in proportion to the increase rates of standard deviation ($c_i$), as indicated in Eq. (12).

$$v_k^{t+1} = v_k^t + N(0, \sigma^t) \tag{11}$$

$$\sigma^{t+1} = \begin{bmatrix} c_d \times \sigma^t, & if & \phi(t) < \delta \\ c_i \times \sigma^t, & if & \phi(t) > \delta \\ \sigma^t, & if & \phi(t) = \delta \end{bmatrix} \tag{12}$$

where, $N(0, \sigma^t)$ : gaussian random variable

$v_k^t$ : $k$-th variable in generation $t$

$\sigma^t$ : standard deviation of the generation $t$

$\phi(t)$ : improved ratio of individual number after adapting mutation operator for population of ES in $t$ generation

$c_d$, $c_i$ : increase and decrease rate of the standard deviation

$\delta$ : constant range from 0 to 1

**(5) Elitism**: The best individual in the population is preserved to perform both GA operations and ES operations in the next generation. This mechanism not only forces GA not to deteriorate temporarily, but also forces ES to exploit information to guide subsequent local searching in the most promising subspace.

The flowchart for searching optimal solutions using the proposed AEA is shown in Fig. 2.

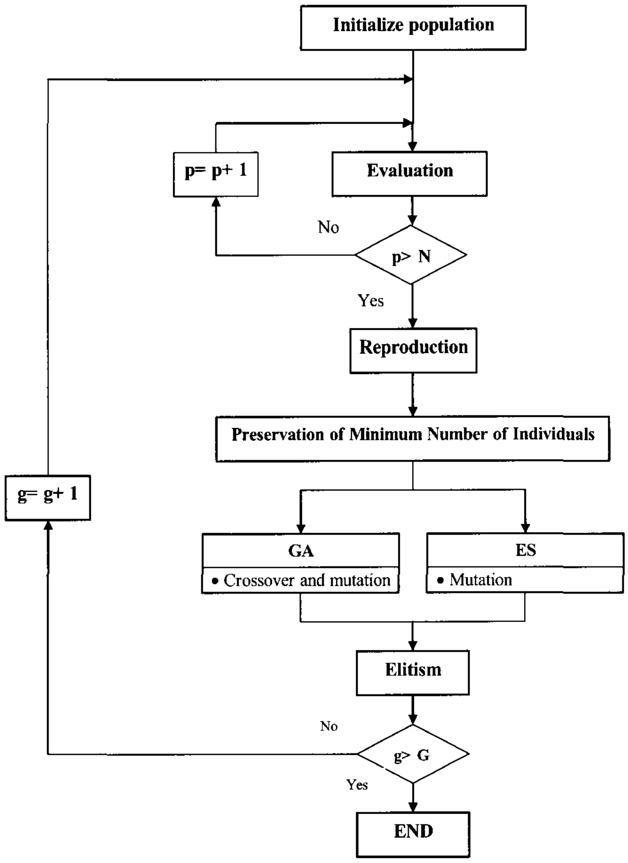



**Fig. 2.** Flowchart for searching optimal solution using AEA

The proposed AEA is paralleled by the PC cluster system to enhance both the solution quality and computation time. Fig. 3 indicates the connection structure between each AEA node. In parallel AEA, AEA operators are executed for each node. Individuals of each AEA node with higher fitness value are transferred to the neighboring AEA node to enhance the search capability of the AEA.

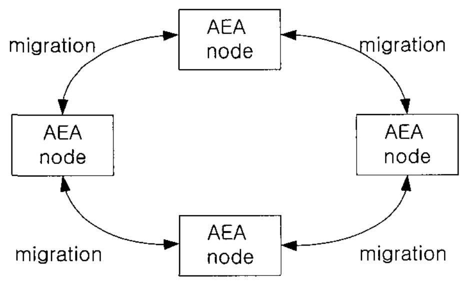



**Fig. 3.** Connection structure of each AEA node

### 3.3 Parallel AEA for the Service Restoration of the Distribution System

To implement the proposed parallel AEA for the service restoration of the distribution systems, we must determine several AEA parameters. The parameters are the ON/OFF switch status, and fitness function considering objectives and constraints, so it is a combinatorial optimization problem.

To solve the service restoration problem by AEA, the initial solution for AEA is randomly selected. AEA for the service restoration encodes the tie switch status, and opens sectionalizing switches in the outage area to transfer and divide load of the outage area to the adjacent tie feeders. If the overload occurs in the tie feeder after restoration, we should decide the tie switch status and open sectionalizing switches in the tie feeder to transfer the parts of the tie feeder load to the adjacent feeders to relieve the overload.

Fig. 4 shows one example of the restoration plans for the outage area. Before a fault, tie switch T1, T2 and T3 were opened and sectionalizing switches S1, S2 and S10 were closed. If we open switch S1 and S2 to clear the fault, the outage area to be restored is represented by the shaded areas. As shown in Fig. 5, tie switch T1 and T3 are determined to be closed. To satisfy the radial constraint, sectionalizing switch S10 is determined to be opened to divide the outage area into 2 parts (Sec. B and Sec D).

Fig. 5 presents one example of the restoration plans for the outage area when the capacity margin of the tie feeder is low. Before a fault, tie switch T1, T2, T3 and T5 were opened and sectionalizing switches S1, S2, S10 and S16 were closed. If we open switch S1 and S2 to clear the fault, the outage area to be restored is represented by the shaded areas. As shown in Fig. 4, tie switch T2 and T3 are determined to be closed. To satisfy the radial constraint, sectionalizing switch S10 is determined to be opened to divide the outage area into 2 parts (Sec. B and Sec. D). After service restoration, tie feeder S/S 2 is overloaded by closing the T2. To relieve the overload of the tie feeder, T5, one of the tie switches is closed and sectionalizing switch S16 is opened considering radial constraint.

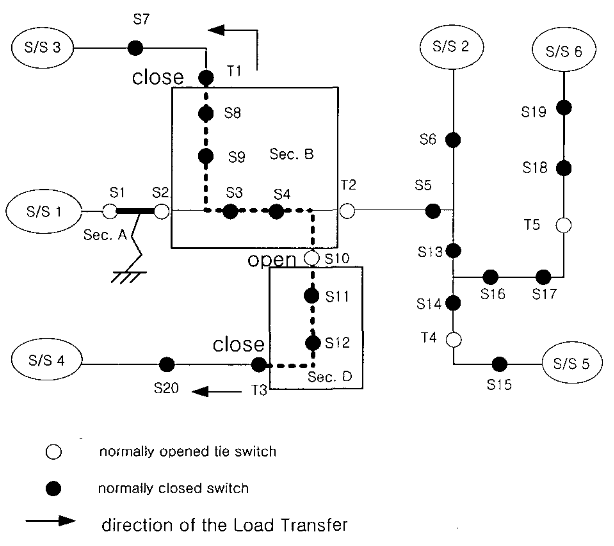



**Fig. 4.** Example for transferring load of outage area to neighboring feeders

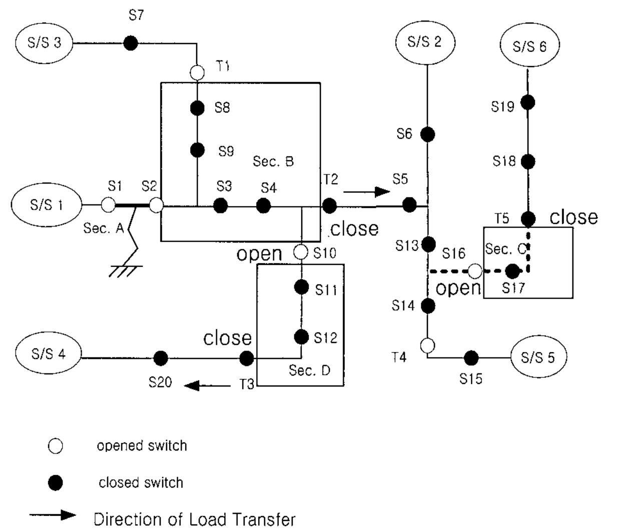



**Fig. 5.** Example for transferring load of outage area to neighboring feeders and transferring load of neighboring feeder to another feeder

With the coding scheme described above, AEA generates an initial population. GA operations, that is, crossover and mutation operations, and ES operation (mutation) are applied to the individuals of the present generations to create the next generation. Fig. 6 represents the string architecture of the AEA-based coding method. As shown in Fig. 6, each string represents the closed tie switches and open sectionalizing switch position of both the outage area and tie feeders with overload after service restoration of the distribution system. For example, suppose one of the strings in AEA is 110: 00010:11111:01010:01:11110:00001 for the example distribution system in Fig. 5. The first 3 bits (110) represent the tie switch status of the outage area. It represents the tie switch T1 and T2 to be closed. To satisfy the radial constraints, among switches between tie switch T1 and T2, we opened S3 by decoding the next consecutive 5 bits (00010). If the overload occurs in the tie feeder of the outage area after load transfer by the service restoration, we should decode the 'tie feeder load transfer' part in the string to determine the open and close switch status in the tie feeder. If the first 2 bits of the 'tie feeder load transfer' section is 01, tie switch T5 is closed. The open switch corresponding to closed tie switch T5 is determined by decoding 5 consecutive bits for sectionalizing switch S18 (00001).

| | Outage Area Load Transfer | | | | | | Tie Feeder Load Transfer | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| String 1 | T1 | T2 | T3 | T1-T2 | T1-T3 | T2-T3 | T4 | T5 | T2-T4 | T2-T5 | * |
| String 2 | T1 | T2 | T3 | T1-T2 | T1-T3 | T2-T3 | T4 | T5 | T2-T4 | T2-T5 | * |
| String n | T1 | T2 | T3 | T1-T2 | T1-T3 | T2-T3 | T4 | T5 | T2-T4 | T2-T5 | * |

where, Ti: i-th switch status (closed:1, opened: 0)
Ti-Tj: opened switch between closed tie switch Ti and Tj
n: no. of population
*: operator code

**Fig. 6.** Coding method of GA for distribution system service restoration

In the AEA evaluation procedures, the fitness of each string can be obtained by the following equations. As shown in Eq. (13), fitness is composed of integral objectives described in Eq. (1) and several constraints such as the line/transformer capacity limit, the voltage-drop limit and the radial constraint.

$$Fitness = \frac{\alpha}{obj + \beta} \tag{13}$$

where, $obj$ : integral objective described in Eq. (1)
$\alpha$ , $\beta$ : constant

In the reproduction procedure, the population of the next generation is selected by roulette wheel selection according to the fitness. Crossover, mutation and elitism are used in the GA operation and Gaussian mutation is used for ES operation.

## 4. Case Studies

To evaluate the effectiveness of the proposed algorithm, service restoration has been conducted on a 22.9kV real-life distribution system in Kangdong in Korea. The system has 7 substations, 17 substation transformers, 100 feeders and 2,558 load sections. The feeder's rated capacity is 10 MVA, and each transformer's capacity is 100MVA. A program for service restoration has been written in C++ and run on an IBM-PC.

Table 2 describes the simulation parameters. Fig. 7 describes the PC cluster system developed in this paper. As indicated in Chapter 2, we restored the faulted section with relevant tie switches for closing to transfer the load of the outage area and opening switch to divide the faulted load relevantly to balance the load of the tie feeder. If overload of the tie feeder occurs after restoration, we can relieve the overload by transferring the part of the load of the tie feeder to the neighboring feeders.

In this paper, we considered the fault restoration problem with less than or equal to 3 outage areas, 2-5 tie switches for each outage area, and about 3 open switches for between closed tie switch pairs. To relieve the overload of the tie feeders, 3-5 switches can be closed and 2-5 switches can be opened for each closed switches to transfer the loads of the overloaded tie feeder to the neighboring feeders. Therefore, the search space of the service restoration problem is approximately 3×3×5×5×3=675. So we changed the population size for 30 to 100 by increasing it by 5 to obtain relevant population size. From these results, we set population size as 80. In selecting the crossover probability, we changed it from 0.6 to 0.9 by increasing it by 0.05. From these results, we set crossover probability as 0.85. Also, we changed the mutation probability from 0.001 to 0.1 by increasing it by 0.005. From these results, we set it as 0.01.

**Table 2.** Simulation coefficients

| Method | AEA | GA | ES |
|---|---|---|---|
| No. of generation | 200 | | |
| No. of population | 80 | | |
| Crossover probability | 0.85 | 0.85 | - |
| Mutation probability | 0.01 | 0.01 | - |
| $c_d$ | 0.985 | - | 0.985 |
| $c_i$ | 1.005 | - | 1.005 |
| $p$ | 0.04 | | |
| $q$ | 0.4 | | |
| $\alpha$ | 500 | | |
| $\beta$ | 0 | | |
| $\alpha_{ILB}$ | 1 | | |
| $\alpha_{IV}$ | 10 | | |
| $p_1, p_2, p_3, p_4$ | 1.3, 5.98, 13485.3, 5.1 | | |

**Fig. 7.** Developed PC cluster system

### 4.1 Service Restoration for an Outage Area

Suppose a permanent fault occurs in section A as indicated in Fig. 8. To clear the fault, switch 23160 and circuit breaker 2419 are tripped. As a result, outage area 1 with a total load of 132.96[A] to be restored is rendered out of service and is connected to 7 feeders through tie switches.

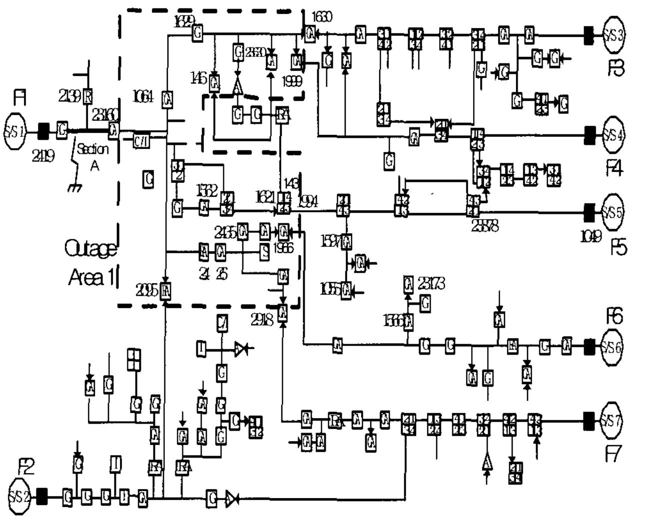



**Fig. 8.** Example distribution system

As described above, the distribution system service restoration problem is one of the combinatorial optimization problems. Therefore, we applied AEA to the distribution system service restoration problem because the proposed algorithm has the global searching capabilities for the problems with objective function with many local minima. Table 3 shows the ranking list of objectives in restoration schemes using the parallel AEA.

**Table 3.** Ranking list of objectives in restoration schemes using the parallel AEA

| Ranking \ Index | $IC^h$ | $ILB^h$ | $IP^h$ | $IV^h$ | Integral objectives |
|---|---|---|---|---|---|
| 1 | 3 | 0.2 | 0 | 0.02 | 5.12 |
| 2 | 3 | 0.2 | 0 | 0.02 | 5.12 |
| 3 | 3 | 0.23 | 0 | 0.02 | 5.3 |

Comparison between the proposed method and the explicit exhaustive method [1] show that the proposed method and the explicit exhaustive method found the same optimal solution. The required switching operations for each ranking are described in Table 4. Table 4 details a ranking list of switching operation sequences using the parallel AEA. The proposed method recommends to the operator that tie switches 1966 and 1630 be closed and sectionalizing switch 24 be opened. This transfer restores service to customers of 132.96[A]. In the above scenario, the load of the tie feeders F3 was 212.05[A], after restoration, which makes the switching index and load balance index 3 and 0.2, respectively. Table 5 describes changes of loads in tie feeders after restoration. The execution time obtained was 18 seconds. This reveals that the proposed algorithm is capable of applying real-time service restoration to a control center operation.

**Table 4.** Ranking list of switching operation sequence

| Ranking | Switching operation sequence |
|---|---|
| 1 | 24 open → 1966 close → 1630 close |
| 2 | 24 open →2918 close → 1630 close |
| 3 | 26 open → 1966 close → 1630 close |

**Table 5.** Feeder load changes after restoration

| Ranking | Feeder | Load changes |
|---|---|---|
| 1 | F3 | 120[A]→212.05 [A] |
|  | F6 | 130[A]→170.91[A] |
| 2 | F3 | 120[A]→212.05 [A] |
|  | F7 | 135[A]→175.91[A] |
| 3 | F3 | 120[A]→215.46[A] |
|  | F6 | 130[A]→167.5[A] |

To reveal the usefulness of the proposed method, its result is compared with that of GA alone and ES alone. Fig. 9 presents the integral objective value according to the generation. As the generation increased, the integral objective value found in each generation decreased to 5.12. The optimal solution is found by GA, ES, and AEA. However, more iterations are needed for GA and ES than AEA to determine the optimal solution.

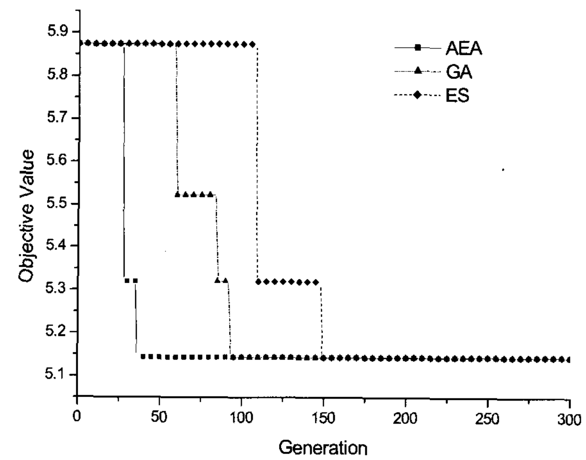



**Fig. 9.** Integral objective value curves for GA, ES, and the proposed parallel AEA

Fig. 10 provides graphs of the number of individuals for GA and ES operation in AEA. As indicated in Fig. 10, the percentage of individuals for GA operation is greater than that of individuals for ES operation in the initial generation. However, from generation to generation, the percentage of individuals for ES operation exceeds that of individuals for GA operation. The AEA produces improved reliability by exploiting the "global" nature of the GA initially as well as the "local" improvement capabilities of the ES from generation to generation.

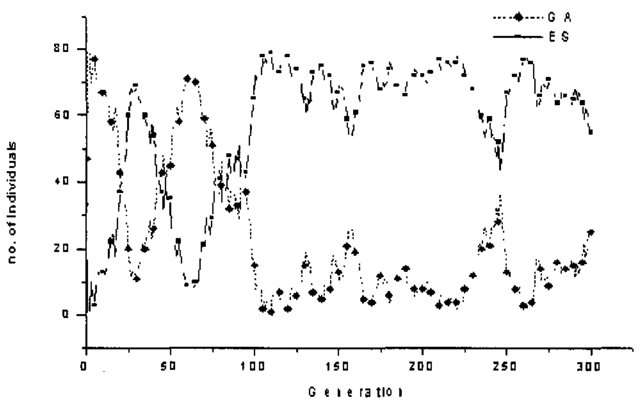



**Fig. 10.** Number of individuals of GA operation and ES operation in each generation

To demonstrate the effects of the parallel characteristics by the PC clustering, speedup and efficiency are evaluated. Speedup and efficiency are described below:

• speedup ( $S_p$ )

$$S_p = \frac{T}{T_p} \tag{14}$$

where, $T$ : run time on one processor

$T_p$ : run time on $p$ processors

• parallel computation efficiency ( $E_p$ )

$$E_p = \frac{S_p}{p} \tag{15}$$

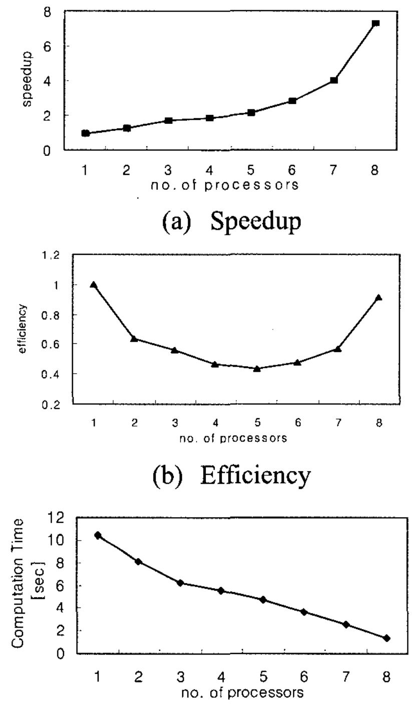
where, $p$ : no. of processors

Fig. 11 presents the speedup, efficiency, and computation time as the number of nodes increases. From Fig. 11, it is found that computation time is decreased while solution quality is maintained. Speedup increased as the number of nodes increased almost linearly, but somewhat lowered because of communication overhead when communication was executed between nodes.

## 4.2 Service Restoration for Several Outage Areas

Suppose a permanent fault occurs of which a section is connected to the branching point as indicated in Fig. 12 (Section B). To clear the fault, switch 23160, 1064, 1562 and 24 are tripped, with 3 outage areas: outage area 1, outage area 2 and outage area 3. As a result, total load of 92.05[A] to be restored is rendered out of service and is connected to 7 feeders through tie switches.

To demonstrate the usefulness of the proposed method, results of the proposed methods are compared with those of GA alone and ES alone. The optimal solution is found by GA, ES and AEA in several iterations because search space in this fault case is small. However, a greater number of iterations are needed for GA and ES than AEA to determine the optimal solution. Also, we cannot obtain the effects of the parallel characteristics. The integral objective value found by GA, ES and AEA decreased to 1.34. Table 6 shows the ranking list of objectives in restoration schemes using the parallel AEA.




(c) Computation time

**Fig. 11.** Speedup, efficiency, and computation time according to the node number

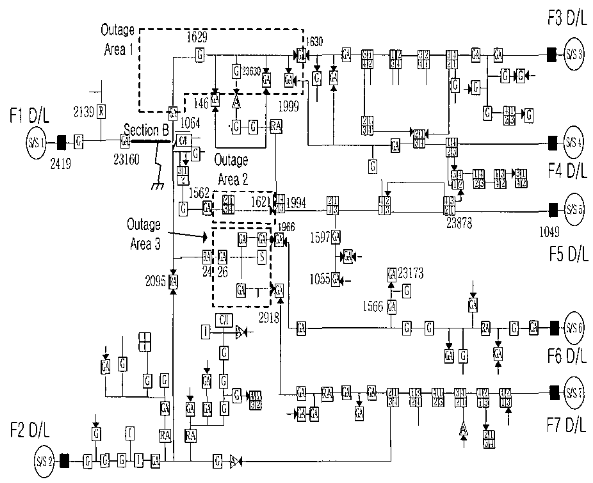



**Fig. 12.** Example distribution system

**Table 6.** Ranking list of objectives in restoration schemes using parallel AEA

| Index / Ranking | $IC^h$ | $ILB^h$ | $IP^h$ | $IV^h$ | Integral objectives |
|---|---|---|---|---|---|
| 1 | 1 | 0.004 | 0 | 0.019 | 1.34 |
| 2 | 1 | 0.004 | 0 | 0.002 | 1.34 |
| 3 | 1 | 0.004 | 0 | 0.002 | 1.34 |

Table 7 describes the ranking list of the switching operation in restoration schemes using the parallel AEA. The proposed method recommends to the operator that tie switches 1630, 1621 and 1966 be closed. This transfer restores service to customers of 92.05[A]. In the above scenario, the load of the tie feeders F3, F5 and F6 were 154.09[A], 152.05[A] and 170.91[A], respectively after restoration, which makes the switching index and load balance index 1 and 0.004, respectively. Table 8 describes changes of loads in the feeders after restoration.

**Table 7.** Ranking list of switching operation sequence

| Ranking | Switching operation sequence |
|---|---|
| 1 | 1630 close → 1621 close → 1966 close |
| 2 | 1999 close → 1621 close → 1966 close |
| 3 | 1630 close → 1621 close → 2918 close |

**Table 8.** Feeder load changes after restoration

| Ranking | Feeder | Load changes |
|---|---|---|
| 1 | F3 | 120[A]→154.09 [A] |
| | F5 | 135[A] →152.05[A] |
| | F6 | 130[A]→170.91[A] |
| 2 | F4 | 125[A]→159.09 [A] |
| | F5 | 135[A]→152.05[A] |
| | F6 | 130[A] → 170.91[A] |
| 3 | F3 | 120[A]→154.09[A] |
| | F5 | 135[A]→152.05[A] |
| | F7 | 135[A]→175.91[A] |

### 4.3 Service Restoration for Multiple Faults

Suppose a permanent fault occurs in the multiple sections (Section A and Section C), as indicated in Fig. 13. To clear the fault, switch 23160, circuit breaker 2419 and switch 23878 and circuit breaker 1049 are tripped. As a result, outage area 1 with a total load of 132.95 [A] to be restored and outage area 2 with a total load of 143.18 [A] to be restored are rendered out of service. Each outage area is connected to 7 feeders through 7 and 2 tie switches, respectively.

Table 9 shows the ranking list of a proposed restoration plan using the parallel AEA. The required switching operations for each ranking are detailed in Table 10. Table 11 describes feeder load changes in tie feeders after restoration. In all ranking, there are several switching operations for transferring load of the backup feeder to relieve the overload of the selected tie feeder to transfer the load of the outage area. In Table 11, backup1 and backup2 signify the feeder name to transfer parts of the load of the tie feeder to reduce the load of the tie feeder.

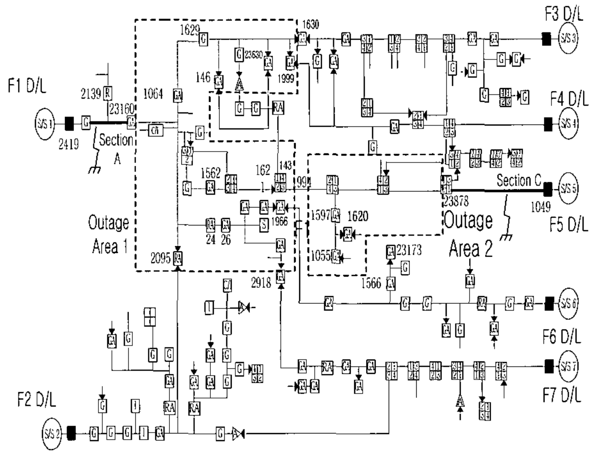



**Fig. 13.** Example distribution system

**Table 9.** Ranking list of objectives in restoration schemes using parallel AEA

| Index / Ranking | $IC^h$ | $ILB^h$ | $IP^h$ | $IV^h$ | Integral objectives |
|---|---|---|---|---|---|
| 1 | 6 | 0.7 | 0 | 0.027 | 11.98 |
| 2 | 7 | 0.58 | 0 | 0.026 | 12.59 |
| 3 | 7 | 0.58 | 0 | 0.026 | 12.59 |

**Table 10.** Ranking list of switching operation sequence

| Ranking | Switching operation sequence |
|---|---|
| 1 | 230 close → 1656 open (transfer load of the backup feeder) → 1630 close →1055 close |
| 2 | 230 close → 1656 open (transfer load of the backup feeder) → 1064 open →1630 close → 2918 close →1055 close |
| 3 | 230 close → 1656 open (transfer load of the backup feeder) → 24 open →2918 close → 2095 close →1055 close |

**Table 11.** Feeder load changes after restoration

| Ranking | Feeder | Load changes |
|---|---|---|
| 1 | F3 | 110[A]→242.96 [A] |
| | Backup 1 | 145[A]→238.34[A] |
| | Backup 2 | 150[A]→199.84[A] |
| 2 | F3 | 110[A]→144.91 [A] |
| | F7 | 135[A]→233.86[A] |
| | Backup 1 | 145[A]→238.34 [A] |
| | Backup 2 | 150[A]→199.84[A] |
| 3 | F3 | 110[A]→208.86[A] |
| | F6 | 130[A]→164.09[A] |
| | Backup 1 | 145[A]→238.34 [A] |
| | Backup 2 | 150[A]→199.84[A] |

To reveal the usefulness of the proposed method, the result of the proposed method is compared to that of GA alone and ES alone. Fig. 14 shows the integral objective index according to the generation. As the generation increased, the integral objective value by GA, ES and AEA found in each generation decreased to 11.98, the optimal solution. However, more iterations are needed for GA and ES than AEA to determine the optimal solution.

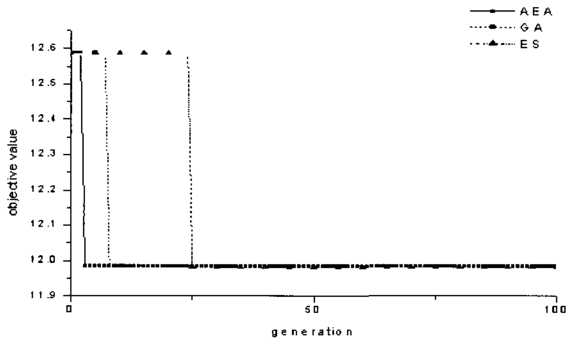



**Fig. 14.** Integral objective value curves for GA, ES and the proposed parallel AEA

Fig. 15 presents graphs of the number of individuals for GA and ES operation in AEA. As indicated in Fig. 15, the percentage of individuals for GA operation is greater than that of individuals for ES operation in the initial generation. However, from generation to generation, the percentage of individuals for ES operation exceeds that of individuals for GA operation. The AEA produces improved reliability by exploiting the "global" nature of the GA initially as well as the "local" improvement capabilities of the ES from generation to generation.

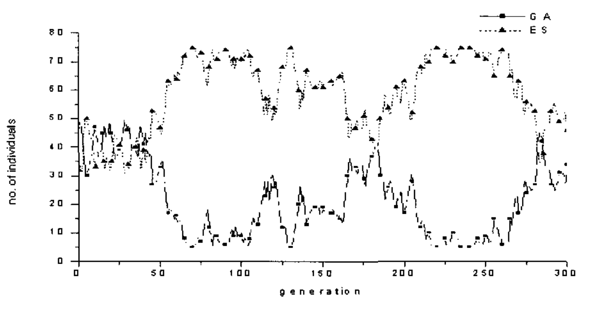



**Fig. 15.** Number of individuals of GA operation and ES operation

Fig. 16 presents the speedup, efficiency, and computation time as the number of nodes increases. From Fig. 16, it is found that computation time is decreased while solution quality is maintained. Speedup increased as the number of nodes increased almost linearly, but somewhat lowered because of communication overhead when communication was executed between nodes.

### 4.4 Service Restoration by Faulted Feeder

Suppose a permanent fault occurs of which section is connected to the faulted feeder as indicated in Fig. 17. To clear the fault, switch 1656 and 2161 are tripped. As a result, a total load of 95.45[A] to be restored is rendered out of service and is connected to 4 tie switches.

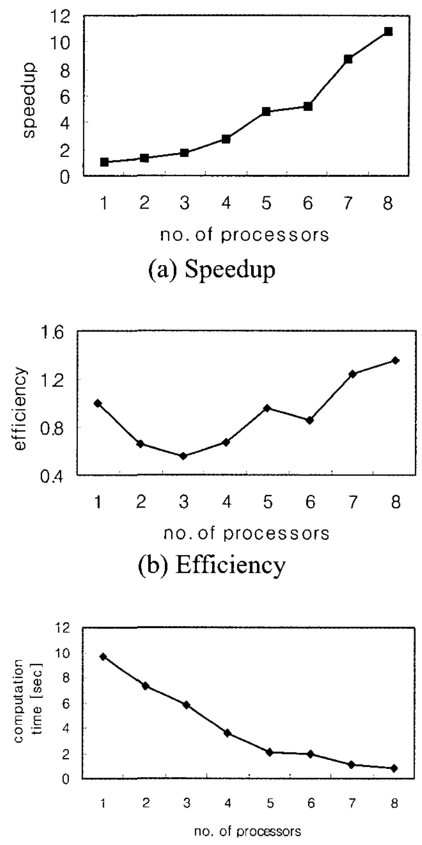



**Fig. 16.** Speedup, efficiency, and computation time according to the node number

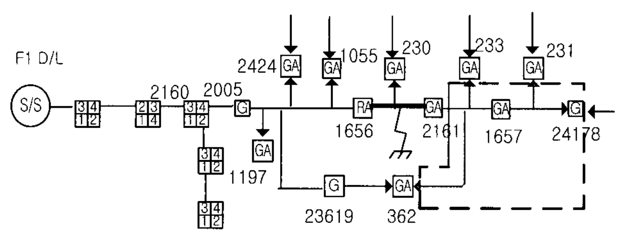



**Fig. 17.** Example Distribution System

To demonstrate the usefulness of the proposed method, its results are compared with those of GA alone and ES alone. The optimal solution is found by GA, ES and AEA. However, more iterations are needed for GA and ES than AEA to determine the optimal solution. Table 12 shows the ranking list of a proposed restoration plan.

**Table 12.** Ranking list of a restoration plan

| Index / Ranking | $IC^h$ | $ILB^h$ | $IP^h$ | $IV^h$ | Integral objectives |
|---|---|---|---|---|---|
| 1 | 0 | 0.0009 | 0 | 0.0002 | 0.006 |
| 2 | 1 | 0.001 | 0 | 0.014 | 1.32 |
| 3 | 1 | 0.002 | 0 | 0.017 | 1.33 |

Table 13 describes the ranking list of switching operation in restoration schemes using the parallel AEA. Table 14 describes feeder load changes after restoration.

**Table 13.** Ranking list of switching operation in restoration schemes using the parallel AEA

| Ranking | Switching operation sequence |
|---|---|
| 1 | 362 close |
| 2 | 233 close |
| 3 | 231 close |

**Table 14.** Feeder load changes after restoration

| Ranking | Feeder | Load changes |
|---|---|---|
| 1 | Faulted D/L | 150[A]→135.94 [A] |
| 2 | Songma D/L | 100[A]→137.5 [A] |
| 3 | Chunsa D/L | 120[A]→157.5[A] |

## 5. Conclusion

This paper presents an application of parallel AEA to search an optimal solution for service restoration of distribution systems. For parallel computing, a PC cluster system consisting of 8 PCs was developed. For compilers of parallel programming, MS visual C++ 6.0 was used under the Windows operating system. In parallel AEA, GA and ES are used in an adaptive manner in order to combine the merits of two different evolutionary algorithms: the global search capability of GA and the local search capability of ES. In the reproduction procedure, proportions of the population by GA and ES are adaptively modulated according to the fitness. The proposed algorithm is paralleled by the PC cluster system to enhance both the solution quality and computation time, and it is very competitive with the parallel machine in terms of cost/ performance.

In this paper, the service restoration problem in distribution systems is formulated as multiple objective problems in order to minimize switch operations and load balance, to satisfy operational constraints such as transformer capacity constraints, feeder loading constraints and voltage drop constraints. To demonstrate the usefulness of the proposed method, it has been applied to a practical distribution system in Korea. As shown in the results, the proposed method has the ability to operate a minimal number of switches, load balancing and to maintain acceptable operating conditions, such as a power flow and system voltage level.

To indicate the usefulness of the proposed method, its results are compared with those of GA alone and ES alone. From the simulation results, it is found that the proposed algorithm is efficient and robust for distribution system reconfiguration in terms of solution quality, speedup, efficiency, and computation time.

## Acknowledgements

This work has been supported in part by EESRI (R-2002-B-044), which is funded by MOCIE (Ministry of Commerce, Industry and Energy).

## References

[1] J. S. Wu, K. L. Tomsovic, C. S. Chen, "A Heuristic Search Approach to Feeder Switching Operations for Overload, Faults, Unbalanced Flow and Maintenance", *IEEE Trans. on Power Delivery*, vol. 6, no. 4, pp. 1579-1585, Oct. 1991.

[2] Whei-Min Lin, Hong-Chan Chin, "A New Approach for Distribution Feeder Reconfiguration for Loss Reduction and Service Restoration," *IEEE Trans. on Power Delivery*, vol. 13, no. 3, pp. 870-875, July 1998

[3] Chen-Ching Liu, Seung Jae Lee, S. S. Venkata, "An Expert System Operational Aid for Restoration and Loss Reduction of Distribution Systems", *IEEE Trans. on Power Systems*, vol. 3, no. 2, pp. 619-626, May 1988.

[4] Shuji Higashi, Jiro Sogawa, Koji Nakata, Akira Manuyama, Akira Hibi, Chihiro Fukui, "Development of the Advanced Intelligent Restoration Guidance System for a Control Center," *International Conference on ISAP*, pp. 731-738, 1994.

[5] Y. Y. Hsu, H. M. Huang, "Distribution System Service Restoration using the Artificial Neural Network Approach and Pattern Recognition Method", *IEE Proceeding Generation Transmission & Distribution*, vol. 142, no. 3, 1995.

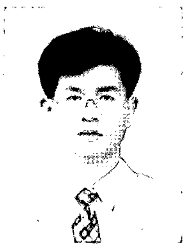
[6] Yuan-Yih Hsu, Han-Ching Kuo, "A Heuristic Based Fuzzy Reasoning Approach for Distribution System Service Restoration", *IEEE Trans. on Power Delivery*, vol. 9, no. 2, pp. 948-953, April 1994.

[7] Seung-Jae Lee, Seong-Il Lim, Bok-Shin Ahn, "Service Restoration of Primary Distribution Systems Based on Fuzzy Evaluation of Multi-Criteria", *IEEE Trans. on Power Systems*, vol. 13, no. 3, pp. 1156-1163, Aug. 1998.

[8] S. H. Chung, K. R. Ryu, S. C. O, T. W. Park, "Parallel

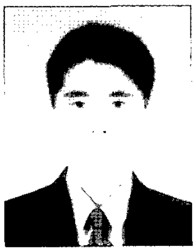
Processing System for High Speed Information Retrieval", *Parallel Processing System Newsletters*, vol. 7, no. 2, pp. 3-19, 1996.
[9] David A. Lifka, "High performance computing with microsoft windows 2000," *Proc. of the 2001 IEEE Conference on Cluster Computing*, pp. 47-54, 2001.
[10] D. E. Goldberg, *Genetic Algorithms in Search, optimization, and Machine Learning*, Addison-Wesley Publishing Company, INC., 1989.
[11] D. B. Fogel, "An Introduction to Simulated Evolutionary Optimization," *IEEE Trans. on Neural Networks*, vol. 5, no. 1, Jan. 1994.
[12] J. Arabas, Z. Michalewicz, and J. Mulawka, "GAVaPS - a Genetic Algorithm with Varying Population Size", *IEEE International Conference on Evolutionary Computation*, pp. 73-78, 1994.
[13] D. Schlierkamp-Voosen and H. Muhlenbein, "Adaptation of Population Sizes by Competing Subpopulations," *IEEE International Conference on Evolutionary Computation*, pp. 330-335, 1996.
[14] D. B. Fogel, L. J. Fogel and J. W. Atmas, "Meta-Evolutionary Programming", *Proceedings 2th Asilomar Conference on Systems, Signals, and Computers*, pp. 540-545, 1991.
[15] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, 1992.

**Kyeong-Jun Mun**
He received his B.S., M.S. and Ph.D. degrees in Electrical Engineering from Pusan National University. He is currently a Senior Researcher at the Korea Atomic Energy Research Institute (KAERI). His research interests are intelligent systems, and power system operation and control, especially related to distribution systems.



**Hwa-Seok Lee**
He received his B.S., M.S. and Ph.D. degrees in Electrical Engineering from Pusan National University in 1991, 1993, and 1997, respectively. He is currently an Associate Professor at Koje College. His research interests include power quality, intelligent systems, and distribution automation.

**June Ho Park**
He received his B.S., M.S. and Ph.D. degrees from Seoul National University of Korea in 1978, 1980, 1987, respectively, all in Electrical Engineering. He is currently a Professor at Pusan National University. His areas of interest are intelligent systems applications to power systems. Dr. Park has been a member of the IEEE Power Engineering Society, Control System Society, and Systems, Man and Cybernetics Society.



**Hyung-Su Kim**
He received his B.S., M.S. and Ph.D. degrees in Electrical Engineering from Pusan National University. He is currently a Post-Doctor at Illinois Institute of Technology. His research interests are intelligent systems, and power system operation and control.

**Gi-Hyun Hwang**
He received his B.S. degree in Electrical Engineering, Kyungsung University and his M.S. and Ph.D. degrees in Electrical Engineering, Pusan National University. He is currently a Professor at Dongseo University and a Manager of the Technology Innovation Center for Ubiquitous Computer Graphics Applications. His research interest lies in applications of intelligent control to power systems, embedded systems, systems on chips, and RFID.