Feeder Reconfiguration Using Binary Coding Particle Swarm Optimization

Wu-Chang Wu and Men-Shen Tsai*

Abstract: This paper proposes an effective approach based on binary coding Particle Swarm Optimization (PSO) to identify the switching operation plan for feeder reconfiguration. The proposed method considers the advantages and disadvantages of existing particle swarm optimization method and redefined the operators of PSO algorithm to fit the application field of distribution systems. Shift operator is proposed to construct the binary coding particle swarm optimization for feeder reconfiguration. A typical distribution system of Taiwan Power Company is used in this paper to demonstrate the effectiveness of the proposed method. The test results show that the proposed method can apply to feeder reconfiguration problems more effectively and stably than existing method.

Keywords: Distribution systems, feeder reconfiguration, particle swarm optimization, shift operator.

1. INTRODUCTION

There are numerous switches on distribution system in general. These switches are divided into two types: sectionalizing-switches (normal closed) and tie-switches (normal open). By changing the on/off status of distribution feeder switches, or feeder reconfiguration, loads can be transferred from one feeder to an adjacent feeder to redistribute loads. Feeder reconfiguration can be used to maintain system balance, reduce feeder losses and improve system reliability.

Many researchers studied the feeder reconfiguration problems using different methods in the past decades. The results of these researches provide acceptable solutions for feeder reconfiguration problems. Heuristic methods to minimize power losses and improve the searching speed were proposed in [1]. Soft computing approaches were also applied to the problem extensively, for example, neural network [2], simulated annealing (SA) [3], genetic algorithm (GA) [4,5] and evolutionary programming (EP) [6,7]. Algorithms based on concept of mimicking swarm intelligent are popular in recent years. For instance, ant colony optimization (ACO) [8-10] and particle swarm optimization (PSO) [11] are the algorithms that can be applied to the field of optimization problems. These algorithms are also applied to the problems of power distribution system gradually.

Kennedy and Eberhart [12,13] proposed an approach called PSO (typical PSO) in 1995. There are many similarities between PSO and Genetic Algorithm (GA). Both algorithms produce an initial solution randomly at first. Through iterations of the evolution process, optimal solution can be obtained. The major difference between GA and PSO is that PSO have no explicit selection, crossover and mutation operations [14]. Searching process in PSO is based on the previous best solution of a particle and the best solution of the population so far to update particle's information. Due to the searching mechanism designed in PSO, the probability of falling into local solution for PSO algorithm can be reduced. Also, PSO is simple and is easy to implement than GA. Thus, PSO is a powerful algorithm to aid and speed up the decision-making process for feeder reconfiguration problem to identify the best switching strategy.

However, the typical PSO is designed for continuous function optimization problems; it is not designed for discrete function optimization problems. Fortunately, Kennedy and Eberhart proposed a modified version of PSO called Binary Particle Swarm Optimization (BPSO) that can be used to solve discrete function optimization problems [15]. In [11], BPSO is used to solve the black-start restoration problems. Although BPSO has been used to solve the problems in the distribution systems, this paper tries to construct a more feasible discrete PSO scheme

Manuscript received February 29, 2008; accepted May 28, 2008. Recommended by Guest Editor Seung Ki Sul. This work was supported by the National Science Council, Republic of China under project contract: NSC 96-2221-E-027-126.

Wu-Chang Wu and Men-Shen Tsai are with Graduate Institute of Automation Technology, National Taipei University of Technology, Taipei, Taiwan. R.O.C. (e-mails: {t5669023, mstsai}@ntut.edu.tw).
* Corresponding author.

based on typical PSO for feeder reconfiguration. The simulation results will compare the proposed method and BPSO to verify the performance and effectiveness.

2. PROBLEM FORMULATION

Feeder reconfiguration is performed by opening/ closing of sectionalizing-switches and tie-switches in distribution systems. The operation of changing on/off status of these switches can reduce line losses or increase the system reliability. The constraints should be considered during configuration. These constraints include: the resulting structure of distribution system must maintain radial structure, feeder capacity should not exceed and feeder voltage profile should be maintained. Since the solution of feeder configuration is the combinations of open/closed switches, the feeder reconfiguration problems can be treated as '1' & '0' permutation combinational optimization problems. '1' represents a normal closed switch and '0' represents a normal open switch. Considering a simple system shown in Fig. 1, the order of switch permutation is sw1, sw2, ..., sw5 in turn. Thus, the status of switch permutation of the system in Fig. 1 is [1 1 0 1 1]. The result of feeder reconfiguration is shown in Fig. 2, and the switch permutation becomes [1 0 1 1 1].

Two objectives are considered in this paper. The first is to minimize the total line losses during normal operation. By doing so, the operation of distribution system will be more economic. The second objective is to distribute loads on feeders evenly. Balanced feeder loads can increase the opportunity of load transfer during emergency conditions. The method proposed in this paper ensures that structure is maintained in radial, ampacity of conductors is kept within allowable limits while minimizing the total line losses and load balancing. "Concentric load model" is used in this paper for calculating branch currents. The line losses can be formulated as follows:

$$F_{loss} = \sum_{i=1}^{n} I_i^2 \cdot z_i, \tag{1}$$

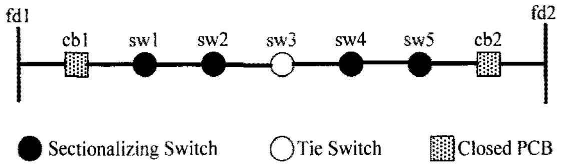


Fig. 1. A simple distribution system.

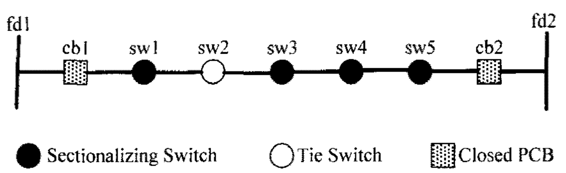


Fig. 2. Result of feeder reconfiguration.

where F_{loss} is the total power losses of distribution feeders, n is the total numbers of zones in distribution system, I_i is the current magnitude of the i-th zone and z_i is the line impendence of the i-th zone. The load balance index is expressed as following:

$$F_{load\_balance} = \sum_{m=1}^{k}\sum_{n=1}^{k}\left(Cap_m - Cap_n\right)^2, \tag{2}$$

where k is number of feeder. Cap_m or Cap_n represents the total load current of a feeder m and n respectively. The total feeder loads can be calculated as following:

$$Cap_i = \sum_j Load_{i,j}, \tag{3}$$

where $Load_{i,j} \in Feeder_i$, i is the feeder number, and j is the load zone number within feeder i.

In order to calculate the fitness value of the system represented by a particle, the method proposed in [7] is used to integrate the two objective functions.

3. PROPOSE APPROACH

3.1. Typical particle swarm optimization

Original concept of PSO came from the study of simulating behavior of bird flocking to look for food. A possible solution for each optimal problem is represented as a particle that is just like a bird flocking in a D-dimensional searching space. Each individual particle has a fitness value evaluated by objective function to pick a good experience for itself and population respectively. PSO initializes particles of population randomly first. Each particle changed its searching direction based on two best values or experiences in each iteration. The first one is the best searching experience of individual so far and is called pbest. The other one is the best result obtained so far by any particle in the population and is called gbest. When pbest and gbest are obtained, a particle updates its velocity and position based on (4) and (5). At last, the algorithm will check the results every iteration until the best solution is found or terminate conditions are satisfied.

$$v_{id}^{new} = wv_{id} + c_1 \times rand() \times (pbest - x_{id}) + c_2 \times rand() \times (gbest - x_{id}), \tag{4}$$

$$x_{id}^{new} = x_{id} + v_{id}^{new}. \tag{5}$$

In the above equations, v_{id} is the original velocity of i-th particle, v_{id}^{new} is the new velocity of i-th particle, w is the inertia weight, c_1 and c_2 are the acceleration constants, x_{id} is the original position of i-th particle, x_{id}^{new} is the new position of i-th particle and rand() is a random number ranging between 0 and 1.

The purpose of updating formula is to lead particles

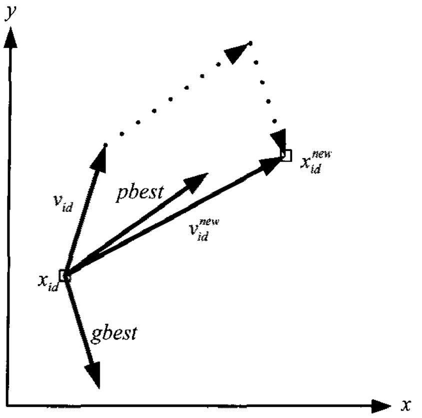


Fig. 3. Searching diagram of typical PSO.

moving toward compound vector of pbest and gbest. By doing so, the opportunity for particle to reach the target (optimal solution) will be increased. In addition, there are interferences in the formula to prevent the algorithm falling into local optimum. The inertia weight in the formula is used to adjust searching areas. A larger inertia weight will motivate the algorithm toward a global search; a smaller one will force the PSO toward a local search. The searching diagram of typical PSO is shown in Fig. 3.

3.2. Binary particle swarm optimization

Kennedy and Eberhart proposed a binary version of PSO for discrete problems [15]. In the binary version, the particle's personal best and global best is still updated as in the typical version. The major difference between binary PSO and typical PSO is that the relevant variables (velocities and positions of the particles) are defined in terms of the changes of probabilities and the particles are formed by integers in {0, 1}. Therefore, a particle flies in a search space restricted to zero and one. The speed of the particle must be constrained to the interval [0, 1]. A logistic sigmoid transformation function $S(v_{id}^{new})$ shown in (6) can be used to limit the speed of particle.

$$S(v_{id}^{new}) = \frac{1}{1+e^{-v_{id}^{new}}}, \tag{6}$$

The update equation of BPSO can be done in two steps. First, (4) is used to update the velocity of the particle and the sigmoid function, (6), is used to limit the velocity in the interval [0, 1]. Second, the new position of the particle is obtained using (7) shown below:

$$\begin{aligned} &if\left(\text{rand}() < S(v_{id}^{new})\right)\ then\ x_{id}^{new} = 1, \\ &else\ x_{id}^{new} = 0, \end{aligned} \tag{7}$$

where rand() is a uniform random number in the range [0, 1].

3.3. Binary coding particle swarm optimization

Through the discussion of typical PSO in the previous section, the PSO algorithm can not be applied to feeder reconfiguration directly. Therefore, the typical PSO must be modified based on the characteristics of distribution feeder operations. Two issues are considered in the modification process. The first one is the problem of feeder reconfiguration is '1' & '0' permutation combinational optimization problem. The second issue is utilizing the shift operator that is used in computer programming languages. This research defines the shift operator and shift operator set using these two aspects. Shift operator and shift operator set can be used to construct the binary coding particle swarm optimization for distribution feeder reconfiguration. These two definitions and the proposed binary coding PSO will be discussed.

3.3.1 Shift operator

Suppose a normal distribution system that has m sectionalizing-switches (normal closed, N.C.) and n tie-switches (normal open, N.O.). The permutation combination of the status of all switches (s=m+n) is [S_1, S_2, ..., S_s] and it will be called 'sequence of switch states', or SSS, in the rest of this paper. The shift operator is defined as SO (Bit_i, $\text{Direction}_{L,R}$, Step_c) and it means that an action will change the position of an N.O. in SSS. Bit_i is the index of i-th switch in SSS. $\text{Direction}_{L,R}$ indicates the direction of left or right shifting on the i-th switch. Step_c is the number of shifting steps. The new permutation in SSS is defined as SSS'=SSS ⟨+⟩ SO. The symbol, '⟨+⟩', represents the shift operator. It applies SO to SSS to get a new SSS'.

A simple example is used to explain the operating process of shift operator. A distribution system shown in Fig. 4 has three feeders, seven N.C.s. and two N.O.es. The SSS of this system is denoted as [1 1 0 1 1 1 0 1 1]. Supposing a SO(3, R, 1) is applied on this SSS. The process of operation is described as Fig. 5.

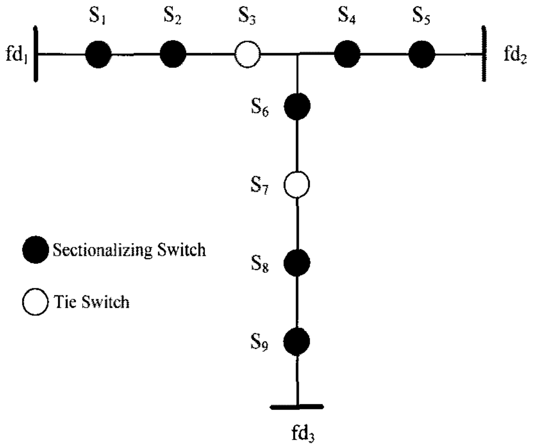


Fig. 4. A simple 3-feeders distribution system.

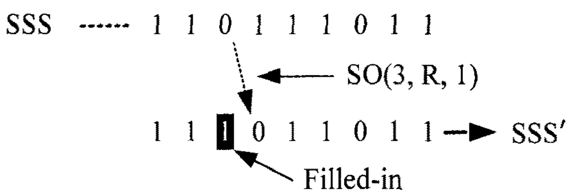


Fig. 5. Basic operating process of shift operator.

When a N.O. shifts, a '1' (N.C.) needs to be filled-in at its original position to maintain system structure.

3.3.2 Shift operator set

A set with at least one or more shift operators is called shift operator set (SOS). An SOS represents all actions about how to fill-in or shift normal open switches on distribution systems. The definition of shift operator set is as below:

$$SOS = \{SO_1, SO_2, ..., SO_n\}, \tag{8}$$

where n is the number of shift operators.

Considering two SSSes, SSS_1 and SSS_2, a set of shift operators which transfers SSS_1 to SSS_2 needs to be identified. Two SSSes, SSS_1=[1 1 0 1 1 1 0 1 1] and SSS_2=[1 0 1 1 1 1 1 0 1], are used to explain how the shift operators are acquired. By comparing the position of normal open switch one by one in these two SSSes, the SOS can be obtained. The determination of the shift operator set and the result are shown as Fig. 6. In this example, SOS={SO_1, SO_2}= $SSS_2 \Theta SSS_1$. The symbol, 'Θ', is used to indicate an action to obtain the shift operators from SSS_1 to SSS_2.

Base on above process, (pbest - x_{id}) and (gbest - x_{id}) in formula (4) can be rewritten as (pbest Θ x_{id}) and (gbest Θ x_{id}) respectively. The x_{id}, pbest and gbest represent different SSSes in this description. This process will transfer an SSS to a new one which is closer to the best switch plan.

3.3.3 Constructing binary coding PSO

The definitions of shift operator and shift operator set are discussed in previous sections. The update formulas (4) and (5) of PSO can be redefined to solve the problem of feeder reconfiguration. The new

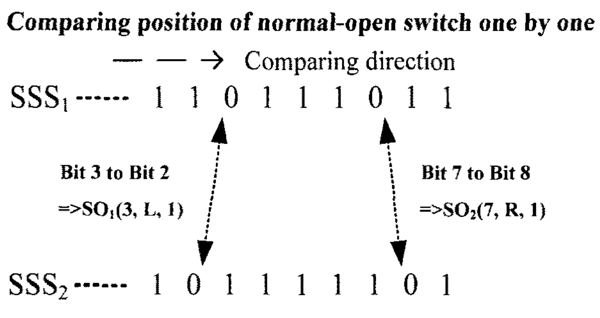


Fig. 6. Decision process of shift operator set.

update formula for the proposed binary coding PSO is as below:

$$\begin{aligned} v_{id}^{new} &= (w \otimes v_{id}) \oplus (rand()\langle\times\rangle(pbest \,\Theta\, x_{id})) \\ &\oplus (rand()\langle\times\rangle(gbest \,\Theta\, x_{id})), \end{aligned} \tag{9}$$

$$x_{id}^{new} = x_{id} \langle + \rangle\, v_{id}^{new}. \tag{10}$$

The symbol, '$\oplus$', shown in (9) is used for combining two shift operator sets. The symbol, '$\otimes$', is the operator that is used to shift the number of steps. The symbol, '$\langle\times\rangle$', is used to select the number of shift operator, SO, in (pbest Θ x_{id}) or (gbest Θ x_{id}) randomly. x_{id} is the original SSS of the i-th particle; pbest is the best SSS of the i-th particle; gbest is the best SSS of any particle in the population. v_{id} is the original shift operator set of the i-th particle, v_{id}^{new} is the new shift operator set of the i-th particle. x_{id}^{new} is the new SSS of the i-th particle. rand() is a random number with a range of [1, n] where n is the number of SO in SOS.

In (9), w is the inertia weight. The role of w is used for adjusting searching areas. The searching areas are reduced progressively when the number of iteration increases. The calculation of inertia weight is shown as (11).

$$w = \frac{iteration_{\max} - iteration_{now}}{iteration_{\max}} \times ShiftStep_{\max} \tag{11}$$

A simple example is used to show how the proposed method works. Based on the system shown in Fig. 4, x_{id}, pbest and gbest represent different SSSes are given below:

x_{id}: [1 1 0 1 1 1 0 1 1],
pbest : [1 0 1 1 1 1 1 0 1],
gbest : [1 1 1 0 1 0 1 1 1].

The SOS can be derived from (pbest Θ x_{id}) and (gbest Θ x_{id}) as:

(pbest Θ x_{id}) = {(3, L, 1), (7, R, 1)},
(gbest Θ x_{id}) = {(3, R, 1), (7, L, 1)}.

The three parts in formula (9) can be expressed as following:

$w \otimes v_{id}$ = {(3, R, 2), (7, R, 2)},
rand() $\langle\times\rangle$ (pbest Θ x_{id}) = {(3, L, 1)},
rand() $\langle\times\rangle$ (gbest Θ x_{id}) = {(7, L, 1)}.

According to (9), the v_{id}^{new} contains four SOes, (3, L, 1), (3, R, 2), (7, L, 1) and (7, R, 2). Combining these four SOes, the final v_{id}^{new} contains two SOes, (3, R, 1) and (7, R, 1). Finally the new SSS, x_{id}^{new}, will be [1 1 1 0 1 1 1 0 1] according to (10).

The procedure of proposed binary coding PSO is summarized as following:

a) Set the size of population and other parameters such as number of iterations and maximum shift steps.
b) Initialize the SSS and shift operator sets randomly to produce particles.
c) Evaluate the fitness value for each particle.
d) Compare the present fitness value of i-th particle with its historical best fitness value. If the present value is better than pbest, update the information including SSS and fitness value of pbest.
e) Compare present fitness value with the best historical fitness value of any particle in population. If the present fitness value is better than gbest, update the information including SSS and fitness value for gbest.
f) Update the shift operation set and generate a new SSS of the particle according to (9) and (10), respectively.
   - If the bit index of an N.O. switch in x_{id}^{new} equals to the index of an N.O. switch in the original x_{id}, then the index of this switch must be shifted right or left few steps randomly.
   - If the bit index of an N.O. switch in x_{id}^{new} after shifting exceeds the bit range, then the index of this switch will be set to the index of an N.O. switch represented in x_{id}, pbest or gbest randomly.
   - If a generated SSS contains fewer numbers of N.O. switches, then the bit indexes of N.C. switches that are in x_{id}, pbest and gbest are chosen and assigned as an N.O. switch. If the revised SSS still has fewer numbers of N.O. switches, then any bit index within the legal range is assigned randomly until the SSS contains enough number of N.O. switch indexes.
g) If stop criterion is satisfied then stop, otherwise go to step c). The stop criterion is the count of iteration reaches the maximum number of iteration.

4. SIMULATION RESULTS

A four-feeder distribution system is used to test the performance of the proposed algorithm. This system is taken from Taoyuan division, Taiwan Power Company, Taiwan. The system has 24 sectionalizing-switches, 8 tie-switches and 28 load-zones, as shown in Fig. 7. The capacity of each feeder is shown in Table 1. The objective functions are: minimizing feeder loss and load balancing index without violating operation constraints. The proposed method and the algorithms described in [15] were implemented using Java language for comparison purposes. Relevant parameters are set as follows. The size of population is 10 for both methods. Maximum number of iteration is set to 1000 for both methods as well. The inertia weight, learning factor of c_1 and c_2 for the methods [15] are set to 0.8, 2.0 and 2.0, respectively. In order to obtain the results and calculate the average performance, 10 runs were performed for each method.

The comparisons of the results from the two algorithms are shown in Table 2. The Max, Min and Average in Table 2 indicate that the maximum, minimum and average losses and load balancing index values in 10 runs respectively. The typical BPSO is not able to get a better result than proposed algorithm due to the higher probability of inadequate number of tie-switches represented by particles. The average values of losses and load balancing index of typical BPSO are higher than proposed method also. The feeders which represent of maximum fitness value of feeder reconfiguration of proposed method and the typical BPSO method are shown in Figs. 8 and 9 respectively. Table 3 shows the results of switch operations for proposed method and typical BPSO. It shows that less switch operations are generated in the proposed method. Table 4 lists the comparison of total

Table 1. Capacity of each feeder.

Feeder ID	F1	F2	F3	F4
Capacity (Amp)	500	500	250	500

Table 2. Results and comparisons of two algorithms.

Method		Binary coding PSO	Typical BPSO [15]
Loss	Max	452kW	478kW
	Min	322kW	326kW
	Average	352kW	370kW
Load Balance Index	Max	351944	531224
	Min	164296	175432
	Average	225105	293192

Table 3. The results of switch operations.

Method	Switch Operation Pair
Binary Coding PSO	{(S4, S31), (S13, S20), (S18, S32)}
Typical BPSO [15]	{(S4, S28), (S6, S7), (S10, S16), (S13, S21)}

Table 4. The comparison of the feeder loading.

Method \ Feeder ID	F1	F2	F3	F4
Original system	176	146	171	203
Proposed Binary Coding PSO	193	170	122	211
Typical BPSO [15]	151	167	110	268

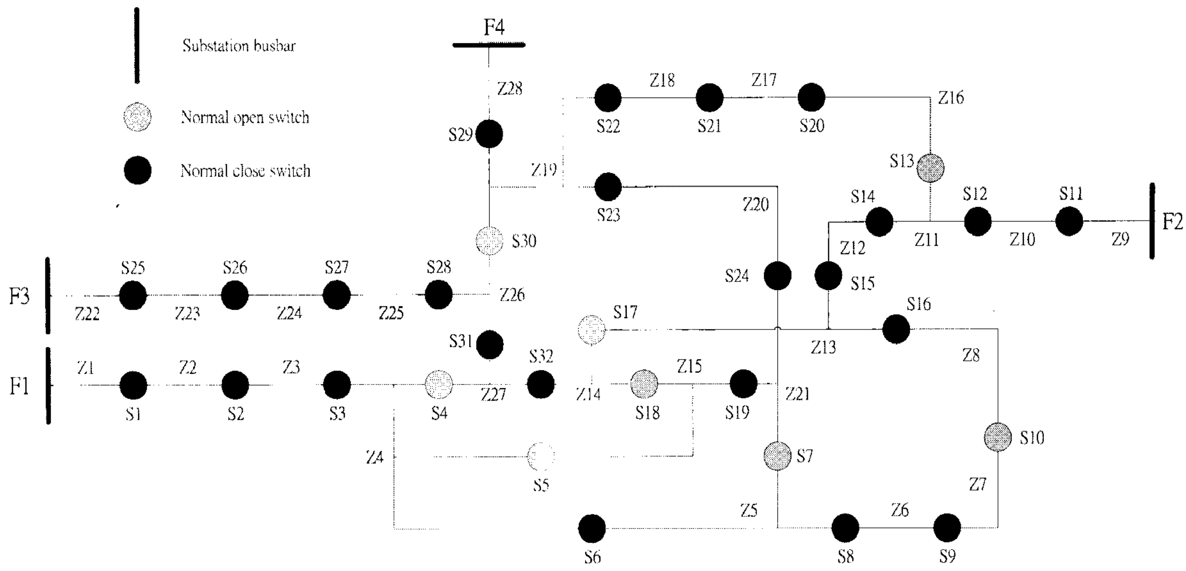


Fig. 7. A four-feeder distribution system for testing.

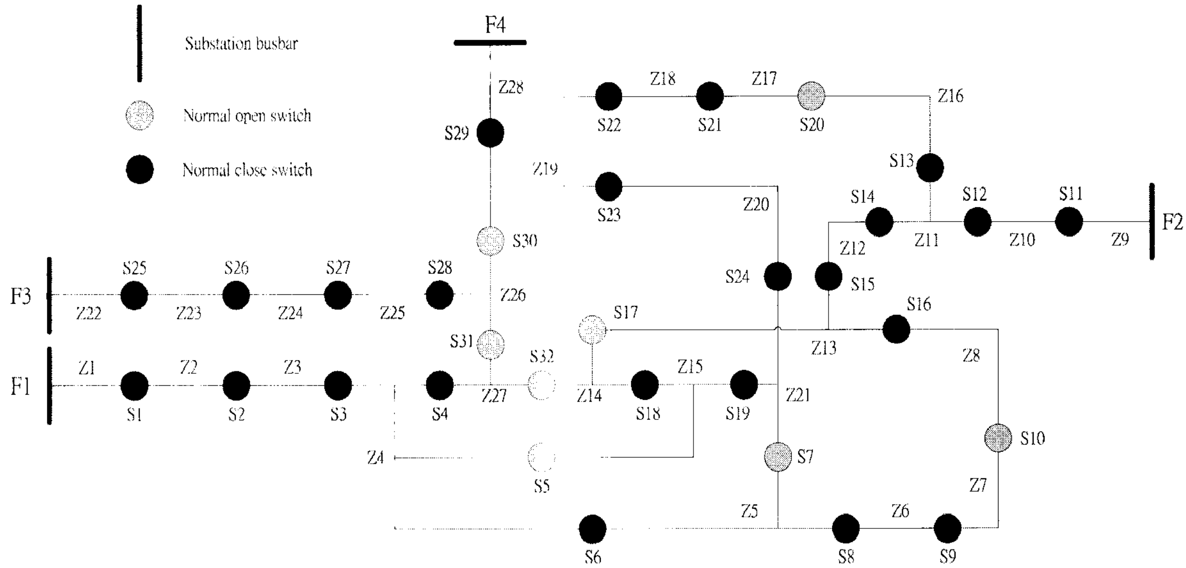


Fig. 8. The final feeder configuration found by the proposed method.

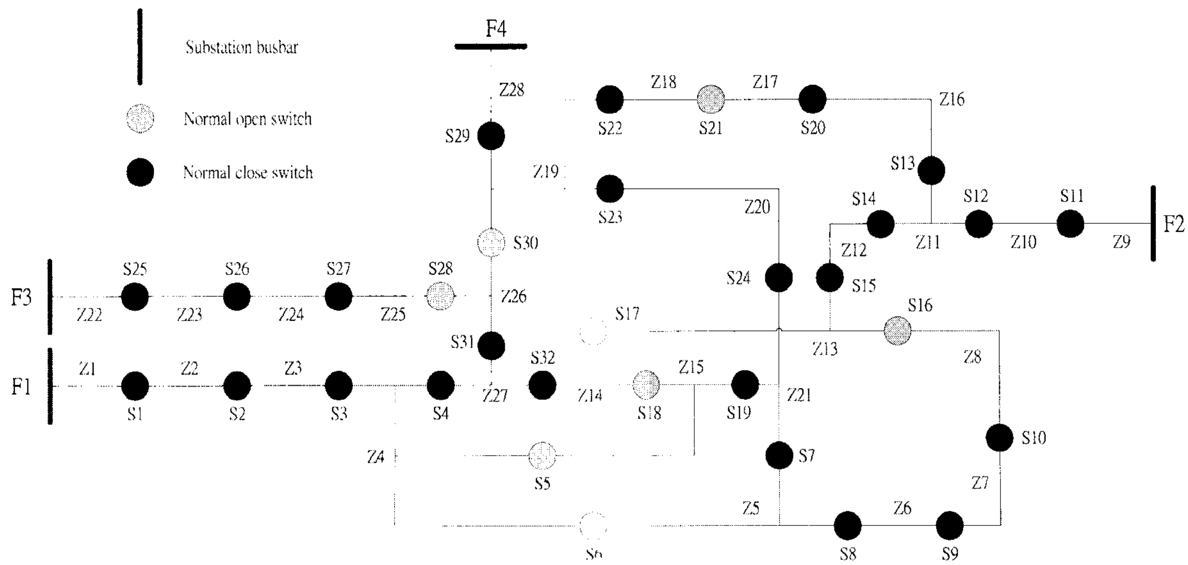


Fig. 9. The final feeder configuration found by the typical BPSO method.

loads of each feeder for original system, the best solution found by the proposed method and the best solution obtained from the typical BPSO. All these results indicate that the proposed method provides better and more reliable solutions than typical BPSO method for minimizing line losses and load balancing problems.

5. CONCLUSIONS

Feeder reconfiguration problems are non-linear discrete optimization problems in nature. Constructing a binary coding particle swarm optimization based on typical PSO to solve this problem is proposed in this paper. In addition, minimizing total line losses and load balancing without violating operation constraints are the objective functions used in this paper. The simulating results show that the proposed method can solve the problem of feeder reconfiguration effectively and stably.

REFERANCE

[1] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Trans. on Power Delivery*, vol. 4, no. 2, pp. 1401-1407, April 1989.

[2] H. Kim, Y. Ko, and K. H. Jung, "Artificial neural networks based feeder reconfiguration for loss reduction in distribution systems," *IEEE Trans. on Power Delivery*, vol. 8, no. 3, pp. 1356-1366, July 1993.

[3] H. C. Chang and C. C. Kuo, "Network reconfiguration in distribution system using simulated annealing," *Electric Power Systems Research*, vol. 29, pp. 227-238, May 1994.

[4] K. Nara, A. Shiose, M. Kitagawa, and T. Ishihara, "Implementation of genetic algorithm for distribution systems loss minimum re-configuration," *IEEE Trans. on Power Systems*, vol. 7, no. 3, pp. 1044-1051, August 1992.

[5] M. Kitayama and K. Matsumoto, "An optimization method for distribution system configuration based on genetic algorithm," *Proc. of IEE APSCOM*, pp. 614-619, 1995.

[6] Y. T. Hsiao, "Mutiobjective evolution programming method for feeder reconfiguration," *IEEE Trans. on Power Systems*, vol. 19, no. 1, pp. 594-599, February 2004.

[7] F.-Y. Hsu and M.-S. Tsai, "A multi-objective evolution programming method for feeder reconfiguration of power distribution system," *Proc. of the 13th Conf. on Intelligent Systems Application to Power Systems*, pp. 55-60, November 2005.

[8] J.-H. Teng and Y.-H. Lui, "A novel ACS-based optimum switch relocation method," *IEEE Trans. on Power Systems*, vol. 18, no. 1, pp. 113-120, February 2003.

[9] E. Carpaneto and G. Chicco, "Ant-colony search-based minimum losses reconfiguration of distribution systems," *Proc. IEEE Melecon*, Dubrovnik, Croatia, pp. 971-974, 2004.

[10] T. Q. D. Khoa and B. T. T. Phan, "Ant colony search based loss minimum for reconfiguration of distribution systems," *Proc. of IEEE Power India Conference*, pages: 6pp, April 2006.

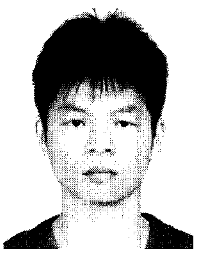
[11] Y. Liu and X. Gu, "Reconfiguration of network skeleton based on discrete particle-swarm optimization for black-start restoration," *IEEE Power Engineering Society General Meeting*, June 2006.

[12] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *Proc. IEEE Int'l. Conf. on Neural Networks, IV*, pp. 1942-1948, 1995.

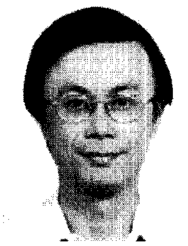
[13] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," *Proc. of IEEE International Conference on Evolutionary Programming*, Alaska, pp. 69-73, May 1998.

[14] R. C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," *Proc. of the 7th Annual Conference on Evolutionary Programming*, San Diego, USA, 1998.

[15] R. C. Eberhart and J. Kennedy, "A discrete binary version of the particle swarm algorithm," *Proc. of IEEE International Conference on Systems*, Man, and Cybernetics, vol. 5, pp. 4104-4108, 1997.



Wu-Chang Wu was born in Taiwan in 1975. In 2002, he received the MS degree from the Department of Electrical Engineering, Chung-Yuan Christian University. He is currently a Ph.D. student at the Graduate Institute of Automation Technology, National Taipei University of Technology. His research areas include applications of expert system and evolutionary techniques to power distribution automation problems.



Men-Shen Tsai was born in Taiwan in 1961. In 1993, he received the Doctoral degree from the Department of Electrical Engineering, University of Washington, Seattle, U.S.A. He is currently an Associate Professor at the Graduate Institute of Automation Technology, National Taipei University of Technology, Taipei, Taiwan. His research areas include applications of intelligent systems to power systems and applications of distributed systems for distribution automation.