

A Novel Binary Coding Particle Swarm Optimization for Feeder Reconfiguration

Men-Shen Tsai and Wu-Chang Wu
National Taipei University of Technology
Taipei, Taiwan

1. Introduction

Power distribution systems are formed by many inter-connected feeders. Each feeder is further partitioned into many load-zones by switches. These switches can be divided into two categories: normally closed sectionalizing-switches and normally opened tie-switches. During normal operation, the structure of distribution system must be maintained in radial structure by properly adjusting the status of the switches. The distribution system can be reconfigured by changing the status of these switches while maintaining the radial structure. The feeder reconfiguration serves several purposes, for example, reducing power losses, maintaining load balance and enhancing service reliability. The mean of a switch operation plan is that by changing the status of sectionalizing-switches and tie-switches, loads can be transferred from one feeder to an adjacent feeder to redistribute loads without violating the operation limitations. However, great deals of switches exist on distribution systems. The number of possible solutions for feeder reconfiguration is increased in exponential order when the number of switches on distribution system increases. Thus selecting the best switch operation plan from all feasible solutions can be considered as an NP-Complete problem. Because the status of switches can be represented as '1' or '0', the problem of feeder reconfiguration can also be regarded as '1' and '0' permutation combinatorial optimization problems.

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 (Baran & Wu, 1989). Soft computing approaches were applied to the problem extensively as well, for example, neural network (Kim et al., 1993), simulated annealing (SA) (Chang & Kuo, 1994), genetic algorithm (GA) (Nara et al., 1992; Kitayama & Matsumoto, 1995) and evolutionary programming (EP) (Hsiao, 2004; Hsu & Tsai, 2005). Algorithms based on concept of mimicking swarm intelligent are popular in recent years. For instance, ant colony optimization (ACO) (Teng & Lui, 2003; Carpaneto & Chicco, 2004; Khoa & Phan, 2006) and particle swarm optimization (PSO) (Chang & Lu, 2002) are the algorithms that can be applied to the field of optimization problems. These algorithms are applied to the problems of power distribution system gradually.

This research will apply the concept of PSO algorithm that is a novel and suitable algorithm for solving combinatorial optimization problems. Kennedy and Eberhart (Kennedy &

Eberhart, 1995; Shi & Eberhart, 1998) proposed PSO (typical PSO) in 1995. The PSO can be treated as the branch of the evolutionary algorithms and it introduces the concept of swarm intelligent. There are many similarities between PSO and Genetic Algorithm (GA). Both algorithms produce an initial solution set randomly at first. Through iterations of the evolution process, optimal solution can be obtained. The major difference between GA and PSO is that PSO has no explicit selection, crossover and mutation operations (Eberhart & Shi, 1998). 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. That means the particles will share the best information between each other and lead the particles moving toward the target. Due to the searching mechanism designed in PSO, the probability of falling into local solution for PSO algorithm can be reduced. Also, the concept of PSO is simple and is easy to implement than GA. Thus, PSO can be a powerful algorithm to aid and speed up the decision-making process for feeder reconfiguration problems to identify the best switching plan.

As mentioned previously, feeder reconfiguration problems are non-linear discrete optimization problems. 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 (Eberhart & Kennedy, 1997). Although BPSO can be applied to solve the discrete optimization problems, there are still problems when BPSO is applied for feeder reconfiguration problems. In feeder reconfiguration problems, there are a large number of tie-switches. Randomly choosing the locations of these tie-switches will cause outages or non-radial structure in distribution systems. In (Chang & Lu, 2002), BPSO is used to solve the feeder reconfiguration problems and the method they proposed avoided the problem of unsuitable numbers of tie-switches. The concept of (Chang & Lu, 2002) is based on BPSO and the moving velocity of particle is defined in terms of probabilities. Instead of BPSO used in (Chang & Lu, 2002), this research tries to construct a more feasible discrete PSO scheme based on typical PSO for feeder reconfiguration. The method proposed in this research modifies the operators of PSO's formula based on the characteristics of both the status of switches and the shift operator to construct the binary coding particle swarm optimization for feeder reconfiguration. Minimizing total line losses and load balancing without violating operation constraints and maintaining radial structure are the two objective functions in this research. The simulations will be performed and the results are used to compare the proposed method, the method proposed in (Chang & Lu, 2002) and BPSO to verify the performance and effectiveness. A distribution system in Taiwan Power Company (TPC) is used in this study to verify the stability and usefulness of the proposed algorithm.

2. Problem Statement

There are all kinds of loads on distribution systems and these loads distributed non-evenly on the distribution feeders. The uneven load distribution on feeders may cause the conductor overloading or transformer load unbalancing on distribution systems during emergency operation. Fig. 1 is a simple 3-feeder distribution system. The ampacity of each feeder is 300A. The total loads on each feeder are 105A, 250A and 200A respectively. This configuration is considered as an unbalanced distribution system when the feeder loading is concerned. The feeder reconfiguration can be performed by opening/closing of

sectionalizing-switches and tie-switches on distribution systems to reduce line losses or increase the system reliability. Therefore, feeder reconfiguration can redistribute the loads and is a common practice for the distribution system operators to avoid the problems of the conductor/transformer overloading or unbalancing on distribution feeders or transformers. Fig. 2 is the result of feeder reconfiguration from Fig. 1. The loads on each feeder are 185A, 190A and 180A respectively after reconfiguration. As a result, the system is operated in a more balanced way. However, some constraints should be considered during feeder reconfiguration. These constraints include: the radial structure of distribution system must be maintained, all zones must be served, feeder capacity should not be exceeded and feeder voltage profile should be maintained. As mentioned earlier, the feeder reconfiguration problems can be treated as '1' & '0' permutation combinatorial optimization problems. '1' represents a normally closed switch; while '0' represents a normally opened switch. Considering a simple system shown in Fig. 1, the order of switch permutation is sw1, sw2, ..., sw11 in turn. Thus, the status of switch permutation of the system in Fig. 1 can be expressed as [1 1 0 1 1 1 1 0 1 1 1]. The result of feeder reconfiguration is shown in Fig. 2, and the switch permutation becomes [1 1 1 0 1 1 1 1 0 1 1].

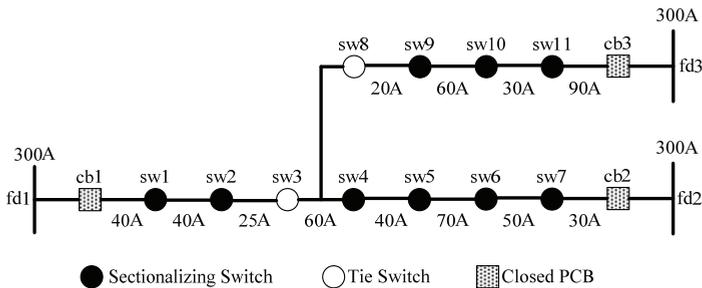


Figure 1. A simple 3-feeders distribution system

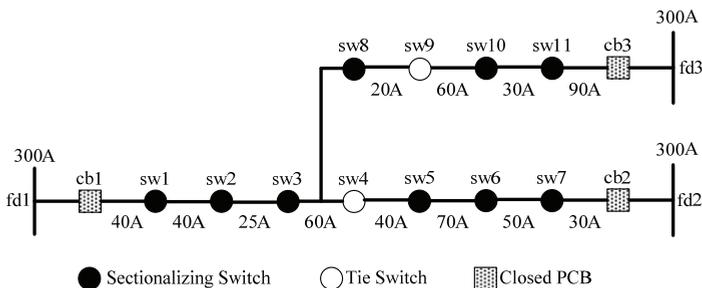


Figure 2. Result of feeder reconfiguration

Some objectives such as minimize the total line losses, minimize the numbers of operating switches, minimize voltage drop and load balance index are considered during feeder reconfiguration in general. Two objectives are considered in this research. The first is to minimize the total line losses during normal operation. By doing so, the operation of distribution system will be more economic and effective. The second objective is to distribute loads on feeders evenly. Balanced feeder loads can increase the opportunity of

load transfer during emergency conditions and improve system reliability. The method proposed in this research also ensures that structure is maintained in radial and the ampacity of each conductor is kept within allowable limits. "Concentric load model" is used in this research for calculating branch currents. The line losses can be formulated as follows:

$$F_{loss} = \text{Re} \left(\sum_{i=1}^n I_i^2 \cdot z_i \right) \quad (1)$$

where F_{loss} is the total real 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 impedance of the i -th zone. The load balance index is expressed as following:

$$F_{load_balance} = \sum_{m=1}^k \sum_{n=1}^k (Cap_m - Cap_n)^2 \quad (2)$$

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

$$Cap_i = \sum_j Load_{i,j} \quad (3)$$

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 (Hsu & Tsai, 2005) is used to integrate the two object functions.

3. Particle Swarm Optimization

3.1 Typical Particle Swarm Optimization

A considerable amount of incredible social behavior and great intelligent exist in nature such as ant colonies, bird flocking, animal herding and fish schooling. Although the ability of individual is limited, the population can achieve the difficult target though cooperation with each other. Note that there is no centralized control in population. The behavior of individual depends on interacting with one another and with their environment only. These simple behaviors among individuals can lead population make themselves toward global behavior. Thus, completing a goal by aggregating the individuals and cooperating with each other that could be called swarm intelligent. Particle Swarm Optimization is one of the optimization algorithms provided with the concept of swarm intelligent. Original concept of PSO came from the study of simulating behavior of bird flocking to look for food. A possible solution for each problem can be represented as a particle that is just like a bird flocking in a D -dimensional searching space. Each individual particle has a fitness value that is evaluated by a fitness function to pick a good experience for itself and population respectively. The particles of population is initialized randomly first. A particle changed its searching direction based on two values or experiences during each iteration. The first one is the best searching experience of individual so far and it is called $pbest$. Another one is the best result obtained so far by any particle in the population and it is called $gbest$. When $pbest$ and $gbest$ are obtained, a particle updates its velocity and position based on (4) and (5). Lastly, the

algorithm will check the results every iteration until the best solution is found or termination conditions are satisfied.

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

$$x_{id}^{new} = x_{id} + v_{id}^{new} \quad (5)$$

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

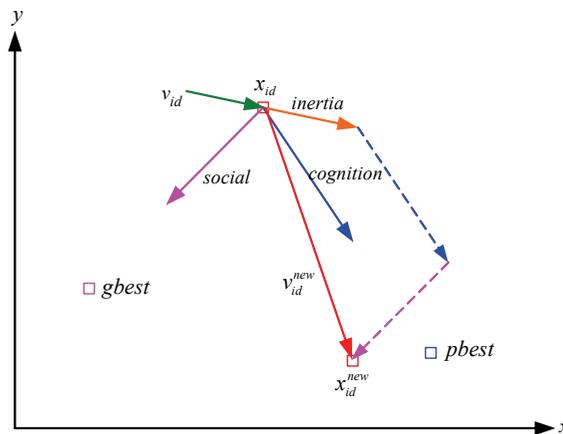


Figure 3. Searching diagram of typical PSO

In (4), the first part is the inertia (habitual behavior), which represents the particle trusts its own status at present location and provides a basic momentum. The second part is the cognition (self-knowledge) or memory, which represents the particle is attracted by its own previous best position and moving toward to it. The third part is the social (social knowledge) or cooperation, which represents the particle is attracted by the best position so far in population and moving toward to it. There are restrictions among these three parts and can be used to determine the major performance of the algorithm. The purpose of updating formula is to lead particles moving toward compound vector of inertia part, cognition part and social part. By doing so, the opportunity for particle to reach the target (optimal solution) will be increased. 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 value 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 (Eberhart & Kennedy, 1997). In the binary PSO version, the particle's personal best and global best is still updated as in the typical version as described in (4). The elements inside x_{id} , $pbest$ and $gbest$

of BPSO are either '1' or '0'. 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.

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

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:

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

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

Since the relevant variables are derived from the changes of probabilities, the concept of BPSO is different from the typical PSO. It is hard to identify the relation between the current status and previous status of a particle. The selection of parameters, such as inertia weight, acceleration constants, etc., is also problematic.

3.3 Binary Coding Particle Swarm Optimization

Through the discussion of typical PSO and BPSO in the previous section, the PSO algorithm cannot be applied to feeder reconfiguration directly. Therefore, this research tries to construct a more feasible discrete PSO scheme based on the concept of typical PSO for feeder reconfiguration. The typical PSO must be modified based on the characteristics of distribution feeder operations. Two issues will be considered in the modification process. The first one is the problem of feeder reconfiguration is '1' & '0' permutation combinatorial optimization problem. The second issue is utilizing the shift operator that is used in computer programming languages. The shift operator and shift operator set defined in this research 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 m sectionalizing switches (normally closed, N.C.) and n tie switches (normally opened, N.O.) exist on a distribution system. The permutation combination of the status all switches ($s=m+n$) is $[S_1, S_2, \dots, S_s]$ and it will be called 'sequence of switch states', or SSS, in the rest of this paper. The shift operator is defined as SO ($Bit_i, Direction_{L,R}, Step_c$) 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. $Direction_{L,R}$ 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 will be applied to SSS to get a new SSS'.

A case is used to explain the operating process of shift operator. A simple distribution system shown in Fig. 4 has four feeders, nine N.C.s and three N.O.es. The SSS of this system is denoted as [1 0 1 0 1 1 1 1 1 0 1]. Supposing an SO(4, R, 1) is applied on this SSS. The process of operation is described as Fig. 5. When an N.O. shifts, a '1' (N.C.) needs to be set at its original position to maintain system structure.

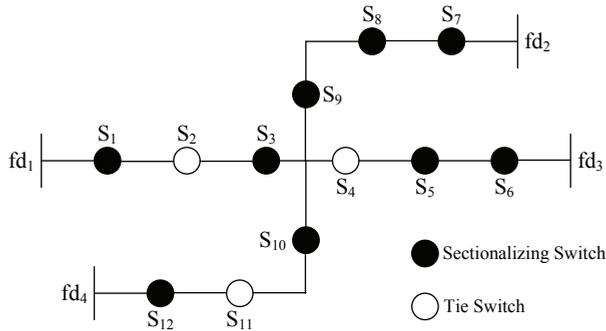


Figure 4. A simple 4-feeders distribution system

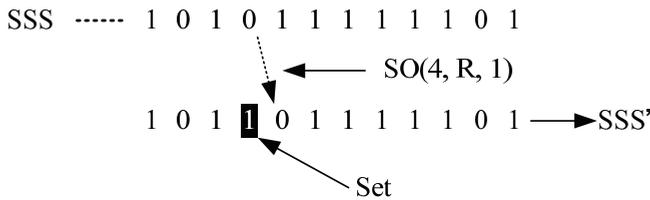


Figure 5. Basic operating process of shift operator

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 set or shift normal open switches on distribution systems. The definition of shift operator set is shown in (8).

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

where n is the number of shift operators.

Considering two SSSes, SSS₁ and SSS₂, a set of shift operators which transfers SSS₁ to SSS₂ needs to be identified. Two SSSes, SSS₁=[1 0 1 0 1 1 1 1 1 0 1] and SSS₂=[1 1 1 0 1 1 0 1 0 1 1], are used to explain how the shift operators are obtained. By comparing the position of normally opened switch one by one in these two SSSes, the SOS can be acquired. The determination of the shift operator set and the result are shown as Fig. 6. In this example, SOS={ SO₁, SO₂, SO₃}= SSS₂ ⊖ SSS₁. The symbol, '⊖', is used to represent an action to get the shift operators from SSS₁ to SSS₂.

Base on the concept of above process, (pbest - x_{id}) and (gbest - x_{id}) in (4) can be rewritten as (pbest ⊖ x_{id}) and (gbest ⊖ x_{id}) respectively. The x_{id}, pbest and gbest represent different

SSSes in this sketch. This process will transfer an SSS to a new one which is closer to the best switch plan.

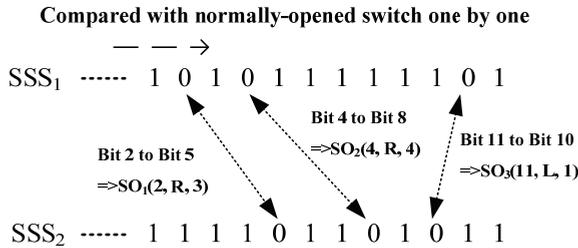


Figure 6. Decision process of shift operator set

3.3.3 Constructing Binary Coding PSO

The definition of shift operator and shift operator set are discussed in previous sections. The velocity update formulas (4) and (5) of PSO can be reestablished to solve the problem of feeder reconfiguration. The new velocity update formula for the proposed binary coding PSO is as below:

$$v_{id}^{new} = (w \otimes v_{id}) \oplus (rand() \langle \times \rangle (pbest \ominus x_{id})) \oplus (rand() \langle \times \rangle (gbest \ominus x_{id})) \quad (9)$$

$$x_{id}^{new} = x_{id} \langle + \rangle v_{id}^{new} \quad (10)$$

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 \ominus x_{id}) or (gbest \ominus 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 Eq. (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 inertia weight can be calculated as (11).

$$w = \frac{iteration_{max} - iteration_{now}}{iteration_{max}} \times ShiftStep_{max} \quad (11)$$

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:

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

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

$$\begin{aligned}(\text{pbest } \Theta_{x_{id}}) &= \{(2, R, 3), (4, R, 4), (11, L, 1)\} \\(\text{gbest } \Theta_{x_{id}}) &= \{(2, R, 1), (4, R, 1), (11, L, 2)\}\end{aligned}$$

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

$$\begin{aligned}w \otimes v_{id} &= \{(2, L, 3), (4, L, 2), (11, R, 2)\} \\ \text{rand}() \langle \times \rangle (\text{pbest } \Theta_{x_{id}}) &= \{(2, R, 3), (4, R, 4), (11, L, 1)\} \\ \text{rand}() \langle \times \rangle (\text{gbest } \Theta_{x_{id}}) &= \{(2, R, 1), (11, L, 2)\}\end{aligned}$$

According to (9), the v_{id}^{new} contains eight SOEs, (2, L, 3), (2, R, 3), (2, R, 1), (4, L, 2), (4, R, 4), (11, R, 2), (11, L, 1) and (11, L, 2). Combining these eight SOEs, the final v_{id}^{new} contains three SOEs, (2, R, 1), (4, R, 2) and (11, L, 1). Finally the new SSS, x_{id}^{new} , will be [1 1 0 1 1 0 1 1 0 1 1] according to (10).

The procedure of proposed binary coding PSO is outlined as below:

- Determine the size of population and other parameters such as number of iterations and maximum shift steps.
- Initialize the SSS and shift operator sets randomly to produce particles.
- Evaluate the fitness value for each particle.
- 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.
- 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.
- Update the shift operator set and generate a new SSS of the particle according to (9) and (10), respectively.
- If stop criterion is satisfied then stop, otherwise go to step c). In this research, the stop criterion is the iteration count reaches the maximum number of iteration.

4. Experimental Results

To verify the performance of the proposed algorithm and compare with algorithms of typical BPSO (Eberhart & Kennedy, 1997) and modified BPSO (Chang & Lu, 2002) for feeder reconfiguration problem, a four-feeder distribution system is used. This distribution 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 (Eberhart & Kennedy, 1997) and (Chang & Lu, 2002) were implemented using Java language for comparison purposes. Relevant parameters are set as follows. The size of population is 10 for all methods. Maximum number of iteration is set to 1000 for all methods as well. The inertia weight, learning factor of c_1 and c_2 for the methods

of typical BPSO (Eberhart & Kennedy, 1997) and modified BPSO (Chang & Lu, 2002) are set to 0.8, 2.0 and 2.0, respectively. The settings of these parameters can be referred to (Chang & Lu, 2002). 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 three algorithms are shown in Table 2. The Max, Min and Average in Table 2 indicate the maximum, minimum and average fitness value, running time, 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. Although the running time of typical BPSO is less than proposed method, the average values of losses and load balancing index of typical BPSO are higher than proposed method. The modified BPSO is able to avoid the problem of inadequate number of tie-switches represented in each particle. On the other hand, the result of proposed method is better than other two methods. Beside the execution time of proposed method is two seconds longer than BPSO, all the other outcomes of proposed method are superior to other methods. The feeders which represent of maximum fitness value of feeder reconfiguration of the typical BPSO method, modified BPSO and proposed method are shown in Fig. 8, Fig. 9 and Fig. 10 respectively. Table 3 lists the comparison of total loads of each feeder obtained from the three methods. All the results indicate that the proposed method provides better and more reliable solutions than typical BPSO and modified BPSO methods for minimizing line losses and load balancing problem.

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

Table 1. Capacity of each feeder

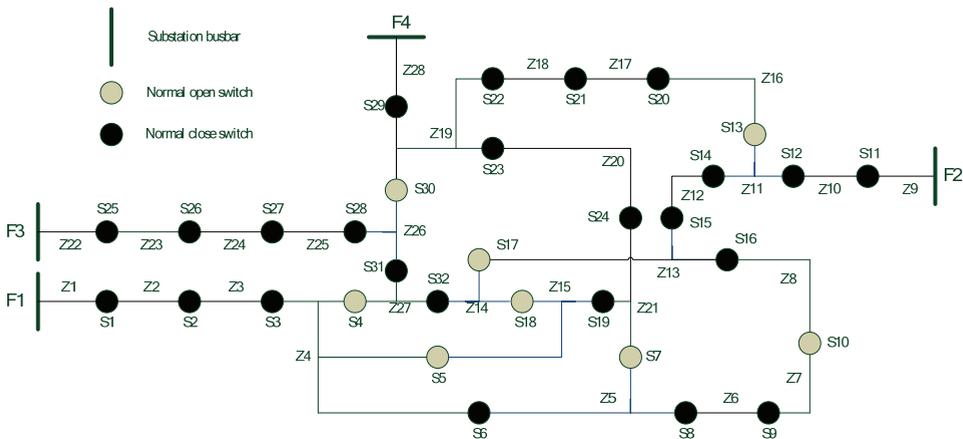


Figure 7. A 4-feeders distribution system for testing

Method		Typical BPSO	Modified BPSO	Binary coding PSO
Fitness Value	Max	0.8759	0.9121	0.9234
	Min	0.8058	0.8844	0.8898
	Average	0.8594	0.8992	0.9032
Running Time (msec)	Max	6625	11015	8734
	Min	5250	8812	8110
	Average	6212	10359	8354
Loss	Max	515kW	405kW	364kW
	Min	339kW	335kW	312kW
	Average	404kW	365kW	329kW
Load Balance Index	Max	525928	434216	264648
	Min	184712	183368	169112
	Average	329504	294859	208328

Table 2. Results and comparisons of three algorithms

Method	Feeder ID			
	F1	F2	F3	F4
Original system	176	146	171	203
Typical BPSO	124	312	122	138
Modified BPSO	139	232	122	203
Binary Coding PSO	139	227	110	220

Table 3. The comparison of the feeder loading

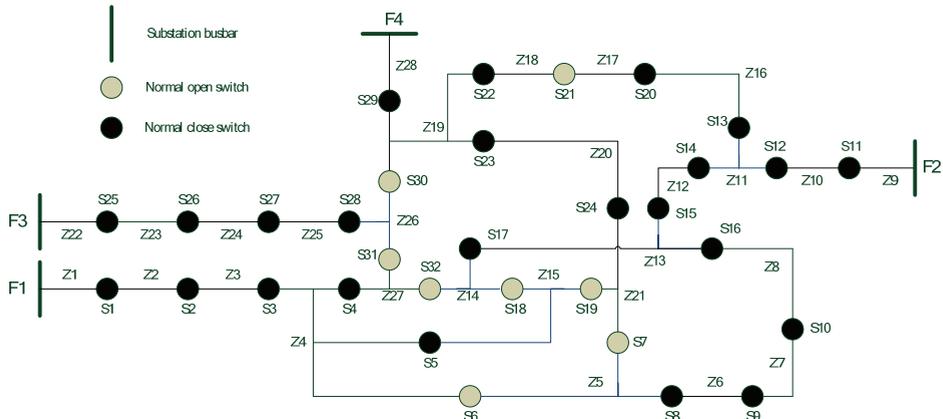


Figure 8. The final feeder configuration found by the typical BPSO method

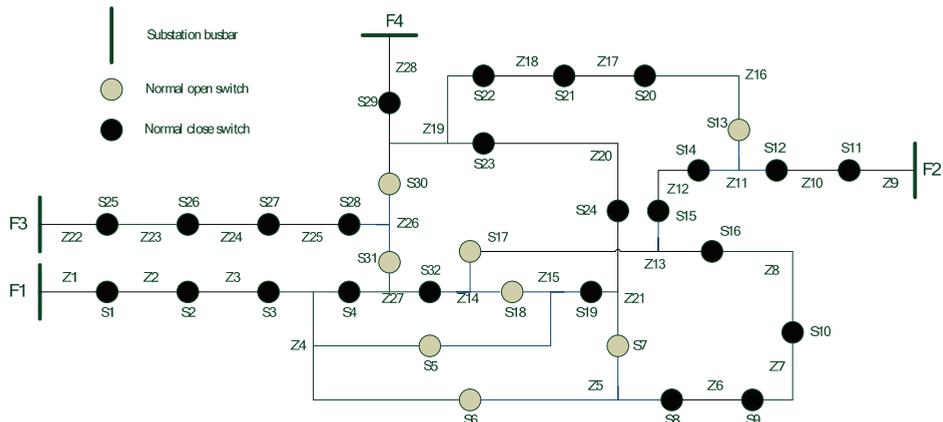


Figure 9. The final feeder configuration found by the modified BPSO method

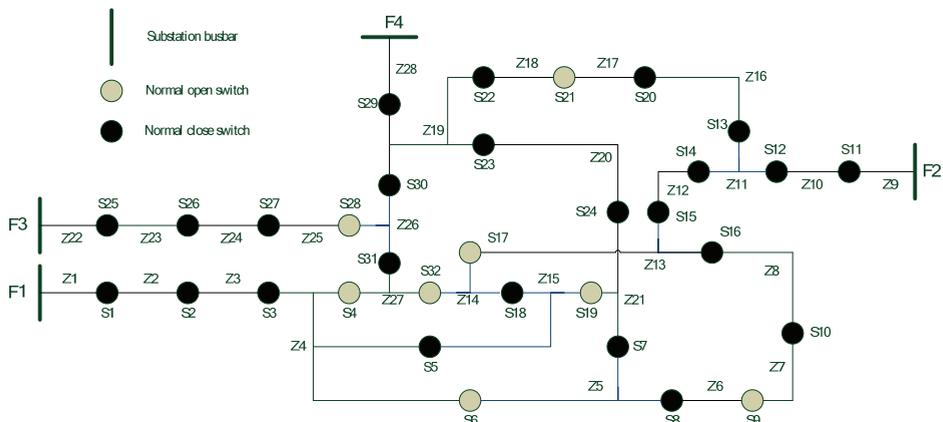


Figure 10. The final feeder configuration found by the proposed method

5. Conclusion

Particle Swarm Optimization is a novel and powerful algorithm for continuous and discrete functions optimization problems. In this work, the concept of typical PSO is modified and applied to the feeder reconfiguration problems. Feeder reconfiguration problems are non-linear discrete optimization problems in nature; and further, there are some defects to use typical BPSO directly for feeder reconfiguration. This research try to construct a binary coding particle swarm optimization based on typical PSO to solve this problem. The operators of typical PSO algorithm have been reviewed and redefined in this research to fit the application of distribution feeder reconfiguration. In addition, minimizing total line losses and load balancing without violating operation constraints are the objective functions used in this research. The experimental results show that the proposed method can solve the feeder reconfiguration problem more effectively.

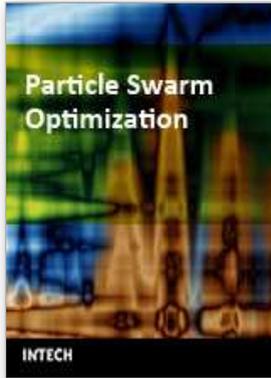
6. Acknowledgement

This research was supported by the National Science Council, Taiwan under contract NSC 97-2221-E-027-110.

7. References

- Baran M.E. and Wu F.F. (1989). Network Reconfiguration in Distribution Systems for Loss Reduction and Load Balancing, *IEEE Trans. on Power Delivery*, vol. 4, no.2, April 1989, pp. 1401-1407.
- Chang H. C. and Kuo C. C. (1994). Network reconfiguration in distribution system using simulated annealing, *Elect. Power Syst. Res*, vol. 29, May 1994, pp. 227-238.
- Chang R.F. and Lu C.N. (2002). Feeder Reconfiguration for Load Factor Improvement, *IEEE Power Engineering Society Winter Meeting*, Vol. 2, 27-31 Jan. 2002, pp.980-984.
- Carpaneto E. and Chicco G. (2004). Ant-Colony Search-Based Minimum Losses Reconfiguration of Distribution Systems, *Proc. IEEE Melecon 2004*, pp.971-974, Dubrovnik, Croatia.
- Eberhart R.C. and Kennedy J. (1997). A Discrete Binary Version of the Particle Swarm Algorithm, *In Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, vol. 5, pp.4104-4108, 1997.
- Eberhart R.C. and Shi Y. (1998). Comparison between Genetic Algorithms and Particle Swarm Optimization, *The 7th Annual Conference on Evolutionary Programming*, San Diego, USA, 1998.
- Hsiao Ying Tung. (2004). Muliobjective Evolution Programming Method for Feeder Reconfiguration, *IEEE Trans. on Power Systems*, Vol. 19, No. 1 pp. 594-599, February 2004.
- Hsu Fu-Yuan and Tsai Men-Shen. (2005). 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.
- Kim H., Ko Y. and Jung K.H. (1993). Artificial Neural Networks Based Feeder Reconfiguration for Loss Reduction in Distribution Systems, *IEEE Trans. on Power Delivery*, vol. 8, no. 3, July 1993, pp. 1356-1366.

- Kennedy J. and Eberhart R. C. (1995). Particle Swarm Optimization, *Proceedings IEEE Int'l. Conf. on Neural Networks, IV*, pp.1942-1948, 1995.
- Kitayama M. and Matsumoto K. (1995). An Optimization Method for Distribution System Configuration Based on Genetic Algorithm, *Proceedings of IEE APSCOM*, pp. 614-619, 1995.
- Khoa T.Q.D. and Phan B.T.T. (2006). Ant Colony Search based loss minimum for reconfiguration of distribution systems, *2006 IEEE Power India Conference*. Page(s): 6pp, April 2006.
- Nara K., Shiose A., Kitagawa M. and Ishihara T. (1992). Implementation of Genetic Algorithm for Distribution Systems Loss Minimum Reconfiguration, *IEEE Trans. on Power Systems*, vol.7, no. 3, August 1992, pp. 1044-1051.
- Shi Y. and Eberhart R.C. (1998). A modified particle swarm optimizer, *IEEE International Conference on Evolutionary Programming*, pp.69-73, May 1998, Alaska.
- Teng Jen-Hao and Lui Yi-Hwa (2003). A Novel ACS-Based Optimum Switch Relocation Method, *IEEE Trans. on Power Systems*, vol. 18, no. 1, February 2003, pp.113-120.



Particle Swarm Optimization

Edited by Aleksandar Lazinica

ISBN 978-953-7619-48-0

Hard cover, 476 pages

Publisher InTech

Published online 01, January, 2009

Published in print edition January, 2009

Particle swarm optimization (PSO) is a population based stochastic optimization technique influenced by the social behavior of bird flocking or fish schooling. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. This book represents the contributions of the top researchers in this field and will serve as a valuable tool for professionals in this interdisciplinary field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Men-Shen Tsai and Wu-Chang Wu (2009). A Novel Binary Coding Particle Swarm Optimization for Feeder Reconfiguration, Particle Swarm Optimization, Aleksandar Lazinica (Ed.), ISBN: 978-953-7619-48-0, InTech, Available from:

http://www.intechopen.com/books/particle_swarm_optimization/a_novel_binary_coding_particle_swarm_optimization_for_feeder_reconfiguration

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.