01 Jan 2006

Particle Swarm Optimization Based Defensive Islanding of Large Scale Power System

Wenxin Liu

Ganesh K. Venayagamoorthy
Missouri University of Science and Technology

David A. Cartes

2006 International Joint Conference on Neural Networks
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Particle Swarm Optimization based Defensive Islanding of Large Scale Power System

Wenxin Liu, *Member, IEEE,* David A. Cartes, *Member, IEEE,*
and Ganesh K. Venayagamoorthy, *Senior Member, IEEE*

***Abstract*— Defensive islanding is an efficient way to avoid catastrophic failures and wide area blackouts. Power system splitting especially for large scale power systems is a combinatorial explosion problem. Thus, it is very difficult to find an optimal solution (if one exists) for large scale power system in real time. This paper proposes to utilize the computational efficiency property of Binary Particle Swarm Optimization (BPSO) to find some efficient splitting solutions in limited timeframe. The solutions are optimized based on a cost function considering the balance between real power generation and consumption, the relative importance of customers, the capacities of distribution and transmission systems, and possibility of region to be impacted, etc. The solutions not only provide the lines to cut but also the corresponding load shedding information in each island. Simulations with large scale power system demonstrate the effectiveness of the proposed algorithm.**

***Index Terms*— Islanding operating, splitting strategies, particle swarm optimization, and system splitting.**

I. Introduction

ALTHOUGH power systems are designed to be tolerant to disturbances, they may become unstable during severe faults, especially when they are operated close to their stability limits. The sources of severe disturbances include earthquakes, hurricanes, human operation errors, control and protection system failures, and malicious attacks, etc.

Studies show that many blackouts could have been avoided and significant losses could have been reduced if proper defensive islanding operations were taken in time prior to or following a catastrophe. Defensive islanding is different from passive islanding. Passive islanding is not under control, and may result from damage and protection. With defensive islanding intentionally deployed to avoid larger losses, the power system will be running in a less versatile, but more robust abnormal state.

Generally, the literature on power system islanding can be classified into two categories. The first category of methods considers the dynamic behavior of power systems, such as normal forms and slow coherency [1-3]. These papers consider the indices of system dynamic behavior and identify the weakest connection. After the grouping of the generators, a brute force search is conducted on the interface network to find the cutsets where the islands are formed. However, this approach has shortcomings, such as increased computational effort and system specific limitations [4]. Furthermore, the solutions usually do not consider the transmission capacity constraint (TCC). The second category of methods tries to split a large scale power system based on its steady state stability [5-6]. To narrow the searching of a strategy space for large scale power systems, the original network is first simplified by graph theory and then OBDDs are used to find the splitting strategies candidates. These methods then check if the candidate solutions satisfy the TCC. The problem with this type of method is that the exact cutting set (the lines to open) cannot be found directly and the method may fail if all of the candidate solutions failed to satisfy the TCC.

For large scale power systems, there may be many splitting strategies that may have similar performance. For practical operating conditions, finding an acceptable solution in real time is much better than finding the optimal solution (if possible) in unacceptable time. Thus, our objective is not to find the optimal solution but some efficient solutions. This paper proposes to use binary particle swarm optimization (BPSO) to find efficient solutions directly from the original power network data. The optimization is based on a cost that is mainly a function of the total working loads in each island. The cost function also gives different indices to different loads to model their vital and nonvital properties. Furthermore, the load shedding information is also obtained during the searching of working loads in each island. During the optimization process, a number of good solutions are recorded as candidate solutions in case the best solution does not satisfy the transmission capacity constraint (TCC). In the worst case, if all of the candidate solutions fail the TCC, the algorithm can avoid these failed solutions by checking with a bad solution repository. Simulation results demonstrate effectiveness of the proposed algorithm.

The rest of the paper is organized as follows. Section II presents some background information on power system islanding and binary particle swarm optimization. Section III presents the detail introduction to the PSO based power system splitting strategy. Section IV gives the some simulation results with different scales of power systems, and finally, some concluding comments in Section V.

This work is supported by the US Department of Energy under Grant No. DE FG02 05CH1292.

Wenxin Liu and Dave A. Cartes are with the Center for Advanced Power Systems (CAPS), Florida State University, Tallahassee, 32310, USA (e-mail: {wliu, dave}@caps.fsu.edu)

Ganesh K. Venayagamoorthy is with Real-Time Power and Intelligent Systems Laboratory, the Department of Electrical and Computer Engineering, University of Missouri - Rolla, MO 65401 USA, (e-mails: ganeshv@umr.edu)

0-7803-9490-9/06/$20.00/©2006 IEEE

II. Background

A. Power System Splitting Problem

There are several important issues to be considered for the power system splitting problem.

At first, the balance between generation and load in islands must be maintained. Otherwise, the islanding operation may result frequency and voltage droop or even blackout within the islands. If the power generation in some island is insufficient to power all loads, the corresponding load shedding information should be accompanied together with the islanding solution.

Since the capacity of transmission and distribution systems are limited, it is necessary to check if they are loaded above their thermal or static stability limits. Thus, any islanding algorithm should be able to provide several candidate solutions, of which to check satisfaction of constraints. If all of the solutions fail, the algorithm should be able to provide some new solutions.

Some more critical customers, such as hospitals, airports, and government buildings, must have higher priority to receive power than other customers. Thus customers should be classified into several classes, such as vital and nonvital, and given different priority indexes. So the performance of some islanding solution should not be judged by the amount of working loads, but should be evaluated by the power supply to the important customers.

For an *n*-line power system, there are 2^n possible solutions to be investigated, which is a combinatorial explosion problem. Thus, it is very difficult to find the optimal solution in real time. Actually, for large scale power system, there may be many solutions that have the same or similar performance. Thus, there is no need to find the only optimal solution (if exist) especially when we need to make a decision in real time. This observation makes it easier to solve the problem. Since timing is an important issue for making online decisions, we should speed up the searching for efficient solutions. To narrow the searching space, we can either simplify the original network first or choose efficient algorithms.

During some predictable events, such as an approaching hurricane, it is better to isolate the possible impacted region from the rest part of the power systems. This kind of isolation can prevent the spreading of possible blackout formed in the impacted region. Furthermore, some dynamic islanding solutions should be prepared for the possible impacted region. That is to say, the impacted region should be split into as many islands as possible so as to minimize losses. This kind of islanding actions does not need to be deployed before hand, but should be taken based on real time operating condition.

After islanding operations, there will be some transients. These oscillations within islands are harmful thus need to be damped out in time. If some islands do not have enough damping ability, this oscillation will exist for a long time and may result bad result. Thus, it is better for the islanding strategy to consider some kind of dynamic response index. Currently, slow coherency has been applied to group the generators according to their parameters. But the slow coherency method requires too much computation and is system specific. Thus, better methods are needed.

B. Binary Particle Swarm Optimization (BPSO)

Kennedy and Eberhart first introduced the PSO algorithm, which is an evolutionary computation technique [9]. The algorithm is derived from the social psychological theory and has been found to be robust for solving problems featuring nonlinearity and nondifferentiability, multiple optima, and high dimensionality through adaptation [10].

Like the other evolutionary computation techniques, PSO is a population based search algorithm and is initialized with a population of random solutions, called particles. Unlike in the other evolutionary computation techniques, each particle in the PSO is also associated with a velocity. Particles fly through the search space with velocities, which are dynamically adjusted according to their historical behaviors. Therefore, the particles have a tendency to fly towards better and better solutions over the course of search process.

The PSO algorithm is simple in concept, easy to implement and computational efficient. The updating rule for PSO algorithm is described as below.

$$\begin{aligned} v_i &= w \cdot v_i + c_1 \cdot rand_1 \cdot (x_p - x_i) + c_2 \cdot rand_2 \cdot (x_g - x_i) \\ x_i &= x_i + v_i \end{aligned} \tag{1}$$

where w, c_1, and c_2 are the inertia weight, cognitive acceleration and social acceleration constants respectively; $rand_1$ and $rand_2$ are two random numbers; x_i represents the location of the ith particle; x_p represents the best solution (fitness) the particle has achieved so far (pbset); x_g represents the overall best location obtained so far by all particles in the population (gbset); v_i represents the velocity of the particle with $v_i^{\min} \le v_i \le v_i^{\max}$.

$v_i^{\max}$ determines the resolution, or fitness , with which regions between the present position and target position are searched. The constants c_1 and c_2 represent the weighting of the stochastic acceleration terms that pull each particle toward pbest and gbest positions. According to past experience, $v_i^{\max}$ is often set at 10-20% of the dynamic range of the variable on each dimension and w, c_1, and c_2 are often set to 0.8, 2, and 2.

The particle swarm works by adjusting trajectories through manipulation of each coordinate of a particle. However, many optimization problems are set in a space featuring discrete, qualitative distinctions between variables and between levels of variables. In the binary version of PSO (BPSO) [11], the trajectories are changes in the probability that a coordinate will take on binary value (0 or 1). For BPSO, the second equation in (1) needs to be modified according to (2).

$$if\ (rand_3 < S(v_i)),\ then\ \mathrm{x}_i = 1,\ else\ \mathrm{x}_i = 0 \tag{2}$$

where $S(v_i)$ is a sigmoid limiting transformation function defined as $S(v) = 1/(1+e^{-v})$, and $rand_3$ is a random number.

III. PSO BASED POWER SYSTEM SPLITTING STRATEGY

The proposed PSO based power system splitting strategy can be described using the following flow chart in Fig. 1. The three stages of the algorithm are described as follows.

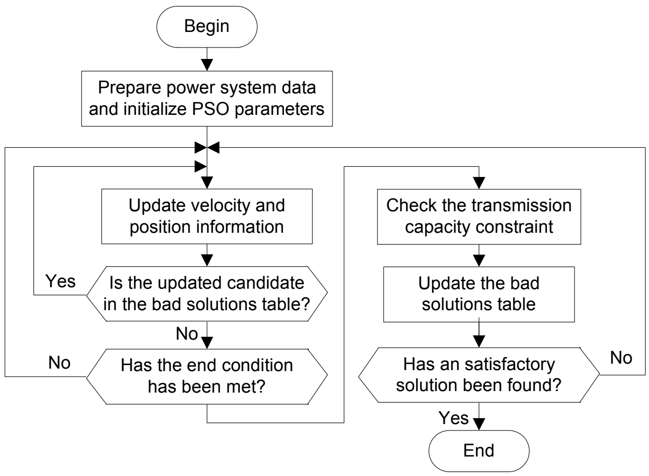

Fig. 1: Flowchart of the PSO based splitting algorithm

A. Preprocessing

The original power network is represented using n_{bus} (the number of buses) bus objects. Fields in each bus object include the buses that it is connected to, the type of the bus (load, generator, etc.), the vital/nonvital property of the bus, the active power generation/consumption, etc. This kind of representation can be easily implemented with those popular object oriented programming languages.

For the previous mentioned predictable events, the possible impacted region should be considered as a whole. Assuming that there are n_{reg} lines in the impacted region, then for a n_{line} line power system, the number of transmission lines to be considered equals n_{line} - n_{reg} + 1, which is the dimension of the PSO particle vector. The elements of the binary vector can be either 1 or 0 to represent the status of the transmission line, '1' means line closed and '0' means line opened.

Other parameters of the PSO algorithm are selected as follows. The population size is selected to be 20, w, c_1 and c_2 are set to 0.8, 2, and 2 respectively, the top 20 solutions will be recorded during optimization, and the optimization process stops after 200 iterations.

B. Optimization

The following cost function is used to investigate the performance of possible solutions.

$$Obj\_fun = \left(\sum_{i=1}^{n} \sum_{j=1}^{m_i} load_{ij} \cdot prior\_idx_j \right) \left(1 + e^{-\frac{(\Delta subcircuit\_num)^2}{a}} \right) \quad (3)$$

where n is the number of subcircuit formed by a investigated solution, m_i is the number of working loads in the ith subcircuit, $load_{ij}$ is the active power consumption of the jth load in the ith subcircuit, and $prior\_idx_j$ is the priority index used to representation the vital/nonvital property of the load, $\Delta subcircuit\_num$ is the difference between the desired and actual subcircuit number for the investigated splitting solution.

To test a candidate solution, the original power system needs to be reconfigured by opening some transmission lines according to the meaning of the solution vector. Then the reconfigured power network will be checked for the number of subcircuits and their included buses. Since a subcircuit is meaningful only when there are both generator(s) and load(s) included, the investigating of the power system structure should begin with some generator until all of the generators have been included in some islands.

During optimization process, a candidate solution will be compared with a bad solutions table and a good solutions table. The bad solutions table includes all known solutions that fail the TCC test and the good solutions table includes top 20 good solutions. If the updated solution is within the bad solution table, then the updated solution is discarded and the updating process is repeated again until it is not included in the bad solution table. If a candidate solution is better than the worst solution in the good solutions table, the good solutions table will be updated.

Because of the random nature of the optimization process, the generated candidate solutions may have the following three bad properties. These random candidate solutions are not good for computational efficiency and thus need to be improved. Firstly, there may be some load buses excluded from all of the islands. But all of the loads should be considered during splitting, unless some loads need to be shed intentionally for generations/loads balance. For this case, the excluded load bus is assigned to an island that it is connected to which has the most generation. Secondly, some intermediate buses that are neither generator buses nor load buses may be excluded. These buses are not important to calculate the generations and loads in subcircuits. The resulting subcircuits have the same generations/loads whether the connected lines are open or close. Since the candidate solutions need to be compared with the good and bad solution tables, the status of the excluded intermediate buses may decrease computational efficiency. For this case, these intermediate buses are connected to islands, which have the least number of buses. Thirdly, according to the randomly generated solutions, some generator bus may not be connected to any loads. This kind of solutions is not practical thus need to be avoided. For this case, the generator buses are connected to the least redundant generation island that it is connected to.

After the configuration of islands has been found, the loads getting power supplied needs to be decided. This problem can be modeled as a bin packing problem in computational complexity theory. The bin packing problem is an NP hard problem. For large scale problems, it is very difficult to find the optimal solution within limited time. To reduce the computation burden for large scale power systems, greedy algorithm is utilized. During this process, the load shedding information can be obtained as a byproduct.

During the optimization process, there may be a lot of different candidate solutions that have the same value of cost function thus need to be sorted. It is assumed that the most preferable solution has the closest number of buses in islands. Thus the following distance is defined for this comparison purpose.

$$dist = \sum_{i=1}^{n} (nob_i - \overline{n}) \tag{4}$$

where n is the number of islands, nob_i is the number of buses in each islands, and $\overline{n}$ is the average value of nob_i.

Since 20 top candidate solutions are recorded in the good solutions table, it does not matter whether there are some solutions with the same values of objective function and the distance.

C. Postprocessing

After the optimization process, all candidate solutions are displayed with the transmission lines to cut, the number of islands to be formed, the buses in each island, the available generation and working loads, and the loads to shed.

During the optimization process, a number of good candidates are recorded. After optimization, the candidate solutions will be checked for the TCC based on power flow calculation. If some candidate solution satisfies the TCC, the searching for optimal splitting strategy ends. Otherwise, the optimization process will start again until a satisfactory solution is obtained.

IV. Simulation Results

The proposed splitting algorithm is tested with three different scale power systems, 9-bus power system, IEEE 30-bus system, and 118-bus power system [7].

A. WSCC 9-bus Power System

The configuration and parameters of the simplified WSCC 9-bus power system are shown in Fig. 2.

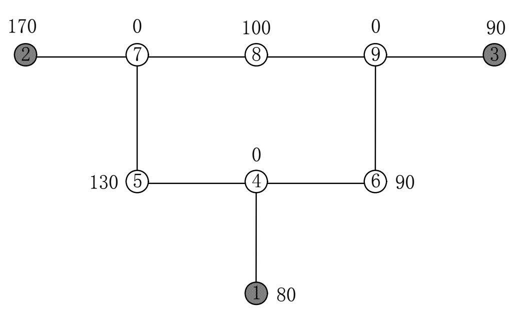

Fig. 2. IEEE WSCC 9-bus power system

In Fig. 2, each circle denotes a bus. Dark circle stands for generator bus, white circle stands for load bus or intermediate bus. The number inside a circle is the index of the bus. The number besides a circle is the active power generation or consumption. The line connecting two buses stands for transmission line. The statuses of the transmission lines decide the formation of islands.

Our objective for this simple example is to find the optimal solution to split the system into two islands. It is easy to see that the optimal solution is to cut two lines, line4-6 and line8-9. Simulation studies show that the algorithm can always find the optimal solution within 5 iterations. That is to say at most $5*20 = 100$ possible solutions were investigated compare with the total $2^9 = 512$ possible splitting solutions.

B. IEEE 30-bus Power System

The configuration of IEEE 30-bus power system is shown in the Fig. 3.

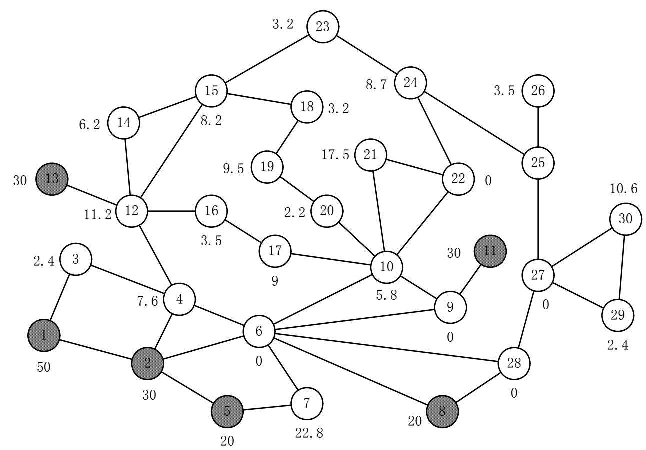

Fig. 3. IEEE 30-bus power system

In the original dataset, the total real power generation is much larger than the total real power loads. To make the problem more challenging, the original data is slightly modified. The modified data is shown in Table I. In the table the bold values are generator buses.

TABLE I
PARAMETERS OF THE WSCC 30-BUS POWER SYSTEM
(TOTAL GENERATION = 180 MW, TOTAL LOADS = 137.5 MW)

Bus No.	P_G/P_L (MW)	Bus No.	P_G/P_L (MW)	Bus No.	P_G/P_L (MW)
1	**50**	**11**	**30**	21	17.5
2	**30**	12	11.2	22	0
3	2.4	**13**	**30**	23	3.2
4	7.6	14	6.2	24	8.7
5	**20**	15	8.2	25	0
6	0	16	3.5	26	3.5
7	22.8	17	9.0	27	0
8	**20**	18	3.2	28	0
9	0	19	9.5	29	2.4
10	5.8	20	2.2	30	10.6

B.1 Split the System into two islands

Top 10 the solutions are shown in Table II. All of these solutions divide the system into two islands with all of loads getting power supply. The solutions are ordered with the increase of distance defined in (4). The numbers outside and inside brackets are the number and the indexes of the buses in the islands respectively.

TABLE II

TOP 10 SOLUTIONS TO SPLIT THE SYSTEM INTO TWO ISLANDS

No.	Islands Info		Opened Lines
	Buses	P_G/P_L (MW)	
1	15 (1, 3~4, 12~19, 23~26)	80/76.2	1-2, 2-4, 4-6, 10-17, 19-20, 22-24, 25-27
	15 (2, 5~11, 20~22, 27~30)	100/61.3	
2	15 (1, 3~4, 12~16, 18~20, 23~26)	80/69.4	1-2, 2-4, 4-6, 10-20, 16-17, 22-24, 25-27
	15 (2, 5~11, 17, 21~22, 27~30)	100/68.1	
3	16 (1~4, 12~16, 18~20, 23~26)	110/69.4	2-5, 2-6, 4-6, 10-20, 16-17, 22-24, 25-27
	14 (5~11, 17, 21~22, 27~30)	70/68.1	
4	14 (1~4, 12~19, 23~24)	110/72.7	2-5, 2-6, 4-6, 10-17, 19-20, 22-24, 24-25
	16 (5~11, 20~22, 25~30)	70/64.8	
5	14 (1, 3~4, 12~16, 18~19, 23~26)	80/67.2	1-2, 2-4, 4-6, 16-17, 19-20, 22-24, 25-27
	16 (2, 5~11, 17, 20~22, 27~30)	100/70.3	
6	16 (1, 3~4, 12~20, 23~26)	80/78.4	1-2, 2-4, 4-6, 10-17, 10-20, 22-24, 25-27
	14 (2, 5, 6~11, 21~22, 27~30)	100/59.1	
7	13 (1, 3~4, 12~20, 23)	80/66.2	1-2, 2-4, 4-6, 10-17, 10-20, 23-24
	17 (2, 5~11, 21~22, 24~30)	100/71.3	
8	17 (1~4, 12~17, 23~27, 29~30)	100/76.5	2-5, 2-6, 4-6, 10-17, 15-18, 22-24, 27-28
	13 (5~11, 18~22, 28)	70/61	
9	12 (1, 3~4, 12~16, 18~20, 23)	80/57.2	1-2, 2-4, 4-6, 10-20, 16-17, 23-24
	18 (2, 5~11, 17, 21~22, 24~30)	100/80.3	
10	11 (1, 3~4, 12~15, 23~26)	80/55	1-2, 2-4, 4-6, 16-17, 19-20, 23-24
	19 (2, 5~11, 16~22, 27~30)	100/82.5	

An example of the optimization process is shown in Fig. 4. In this figure, the total working load is plotted versus iterations. The objective function is not plotted because it is the direct reflection of total working loads. Even though the total working loads converge to the total loads of the system after 10 iterations, it is good for the optimization process to run for longer time so as to find other good solutions that have the same objective functions.

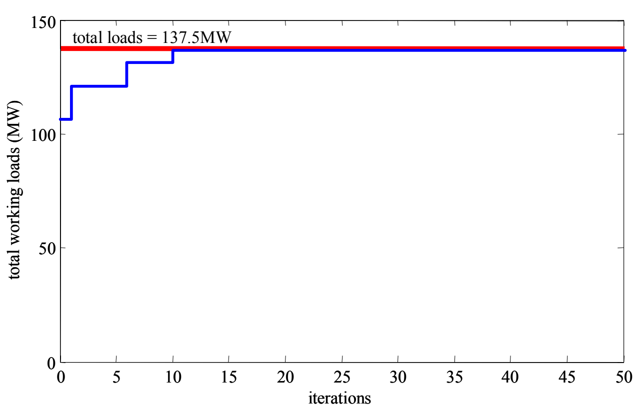

Fig. 4. Optimization process for the IEEE 30-bus power system

Simulation studies show that the algorithm usually converges within 20 iterations, that is to say only 400 possible solutions are investigated. The simulation time on a Pentium IV 3.6G Hz CPU with 1G memory PC is only 6 seconds. Considering the $2^{41} = 2.199*10^{12}$ possible solutions for the 41-line power system, it can be seen that the algorithm is very computationally efficient.

B.2 Split the System into three islands

In this part, the desired number of islands is set to 3. After 200 iterations, the algorithm gives 20 good results that are shown in Table III.

TABLE III

TOP 10 SOLUTIONS TO SPLIT THE SYSTEM INTO TWO ISLANDS

No.	Islands Info		Opened Lines
	Buses	P_G/P_L (MW)	
1	11 (1, 3~4, 12~16, 18~20)	80/54	1-2, 2-4, 4-6, 6-8, 6-28, 10-20, 15-23, 16-17, 25-27
	14 (2, 5~7, 9~11, 17, 21~26)	80/70.5	
	5 (8, 27~30)	20/13	
2	12 (1, 3~4, 12~16, 23~26)	80/54.5	1-2, 2-4, 4-6, 6-8, 6-28, 15-18, 16-17, 22-24, 25-27
	13 (2, 5~7, 9~11, 17~22)	80/70	
	5 (8, 27~30)	20/13	
3	14 (1, 3~4, 12~16, 18~19, 23~26)	80/67.2	1-2, 2-4, 4-6, 6-8, 6-28, 16-17, 19-20, 22-24, 25-27
	11 (2, 5~7, 9~11, 17, 20, 21~22)	80/57.3	
	5 (8, 27~30)	20/13	
4	12 (1, 3~4, 12~15, 22~26)	80/51	1-2, 2-4, 4-6, 6-8, 6-28, 10-22, 12-16, 15-18, 21-22,25-27
	13 (2, 5~7, 9~11, 16 ~21)	80/73.5	
	5 (8, 27~30)	20/13	
5	15 (1, 3~4, 12~19, 23~26)	80/76.2	1-2, 2-4, 2-6, 4-6, 6-7, 10-17, 19-20, 22-24, 25-27
	3 (2, 5, 7)	50/22.8	
	12 (6, 8~11, 20~22, 27~30)	50/38.5	
6	13 (1, 3~4, 12~16, 18~20, 23~24)	80/65.9	1-2, 2-4, 2-6, 4-6, 6-7, 10-20, 16-17, 22-24, 24-25
	3 (2, 5, 7)	50/22.9	
	14 (6, 8~11, 17, 21~22, 25~30)	50/48.8	
7	14 (1, 3~4, 12~16, 18~19, 23~26)	80/67.2	1-2, 2-4, 2-6, 4-6, 6-7, 16-17, 19-20, 22-24, 25-27
	3 (2, 5, 7)	50/22.8	
	13 (6, 8~11, 17, 20~22, 27~30)	50/47.5	
8	17 (1, 3~4, 12~16, 18, 23~30)	80/70.7	1-2, 2-4, 2-6, 4-6, 6-7, 6-28, 16-17, 18-19, 22-24
	3 (2, 5, 7)	50/22.8	
	10 (6, 8~11, 17, 19~22)	50/44	
9	20 (1~7, 9~11, 15~24)	130/103.6	4-12, 6-8, 6-28, 12-15, 12-16, 14-15, 24-25
	7 (8, 25~30)	20/16.5	
	3 (12~14)	30/17.4	
10	3 (1, 3, 4)	50/10	1-2, 2-4, 4-6, 4-12, 10-17, 12-15, 14-15
	22 (2, 5~11, 15, 18~30)	100/97.6	
	5 (12~14, 16~17)	30/29.9	

B.3 Split Tight System into two islands

For the previous two examples, the total generations (180 MW) are larger than that of the total loads (137.5 MW), thus load shedding is not necessary for the top solutions. To test the algorithm's performance for tight system and to demonstrate that the algorithm is able to give related load shedding information, the generations of bus 1 and 2 are both reduced to 10 such that the total generation is 120MW, which is smaller that the total loads. The optimization process stops after the algorithm is run for 200 iterations. The 20 good solutions in the good solution table are shown in Table IV.

TABLE IV
TOP 10 SOLUTIONS TO SPLIT THE SYSTEM INTO THREE ISLANDS

No.	Islands Info		Opened Lines	Working load	Loads to shed
	Buses	P_G/P_L (MW)			
1	20 (1~8, 12~16, 18~20, 27~30)	90/89.8	6-9, 6-10, 10-20, 15-23, 16-17, 25-27	119.5	9, 13, 23, 25
	10 (9~11, 17, 21~26)	30/47			
2	3 (1, 3, 4)	10/10	1-2, 2-4, 4-6, 4-12,	119.5	15, 17, 19, 22, 25, 28
	27 (2, 5~30)	110/127.5			
3	17 (1~8, 21~22, 24~30)	60/75.5	4-12, 6-9, 6-10, 10-21, 10-22, 23-24	119.4	2, 3, 13, 17, 22, 25
	13 (9~20, 23)	60/62			
4	8 (1, 3~4, 12~14, 16~17)	40/39.9	1-2, 2-4, 4-6, 10-17, 12-15,14-15	119.4	9, 17, 22, 25
	22 (2, 5~11, 15, 18~30)	80/97.6			
5	25 (1~11, 15, 18~30)	90/107.6	4-12, 10-17, 12-15, 14-15	119.4	9, 17, 22, 25, 28
	5 (12~14, 16~17)	30/29.9			
6	21 (1~2, 5~11, 17, 19~22, 24~30)	90/92	1-3, 2-4, 4-6, 16-17, 18-19, 23-24	119.2	13, 15, 17, 22
	9 (3, 4, 12~16, 18, 23)	30/45.5			
7	15 (1~5, 7, 12~18, 23~24)	70/86	2-6, 4-6, 6-7, 10-17, 18-19, 22-24, 24-25	119.2	13, 15, 17, 22
	15 (6, 8~11, 19~22, 25~30)	50/51.5			
8	23 (1~8, 12~16, 21~30)	90/107.8	6-9, 6-10, 10-21, 15-18, 16-17	118.7	13, 15, 22, 25, 28
	7 (9~11, 17 ~20)	30/29.7			
9	22 (1~3, 5~11, 16~17, 21~30)	90/89.4	2-4, 3-4, 4-6, 10-20, 12-16, 15-23	118.3	3, 13, 17, 25
	8 (4, 12~15, 18~20)	30/48.1			
10	22 (1~8, 12~18, 23, 25~30)	90/93.8	6-9, 6-10, 10-17, 18-19, 23-24, 24-25	118.2	2, 3, 9, 23, 28
	8 (9~11, 19~22, 24)	30/43.7			

C. *IEEE 118-bus power system*

The data of the IEEE 118-bus power system is shown in Table V. There are 15 generator buses, 93 load buses and 10 intermediate buses.

C.1 Split a Large Scale Power System into Two Islands

The objective of the test is to test the performance of the proposed algorithm in splitting a large scale power system into two islands. The algorithm is run for 2000 iterations and the top 10 good solutions are given in Table IV. The simulation time for the 2000 iteration is approximate five minutes using the above mentioned computer. All of these solutions can divide the system into two islands and all of the loads can get power supply.

C.2 Islanding of the Possible Impacted Region

To isolate a Possible Impacted Region (PIR), we need first find the generations and loads in that region. The PIR will be represented as one bus called PIR bus. In PIR, if the total generation is larger, then the PIR bus is modeled as generator bus, if the total load is larger, then the PIR bus is modeled as a load bus. During the optimization process, the first criteria is still the objective function defined in (3), the second criteria is now changed to the amount of loads in the island where the PIR bus resides. After the exact cutting set is decided, the islanded PIR may be further divided into smaller and smaller islands until the objectives has been met. Since the simulation results will looks similar to the previous examples, simulations results for this case are omitted.

TABLE V
UNITS FOR MAGNETIC PROPERTIES
(TOTAL GENERATION = 3785.4 MW, TOTAL LOADS = 3632 MW)

Bus No.	P_G/P_L (MW)	Bus No.	P_G/P_L (MW)	Bus No.	P_G/P_L (MW)
1	51	41	37	81	0
2	20	42	96	82	54
3	39	43	18	83	20
4	21	44	16	84	11
5	0	45	53	85	24
6	52	46	9	86	21
7	19	47	34	**87**	**4**
8	28	48	20	88	48
9	0	**49**	**117**	**89**	**607**
10	**450**	50	17	90	163
11	70	51	17	91	10
12	**38**	52	18	92	65
13	34	53	23	93	12
14	14	54	65	94	30
15	90	55	63	95	42
16	25	56	84	96	215
17	11	57	12	97	22
18	60	58	12	98	5
19	45	59	122	99	17
20	18	60	78	**100**	**28**
21	14	**61**	**160**	101	5
22	10	62	77	102	17
23	7	63	0	**103**	**38**
24	13	64	0	104	3
25	**220**	**65**	**391**	105	1
26	**314**	**66**	**353**	106	43
27	71	67	28	107	50
28	17	68	0	108	2
29	24	**69**	**516.4**	109	8
30	0	70	66	110	39
31	36	71	0	**111**	**36**
32	59	72	12	112	68
33	23	73	6	113	6
34	59	74	68	114	8
35	33	75	47	115	22
36	31	76	68	116	184
37	0	77	61	117	20
38	0	78	71	118	33
39	27	79	39		
40	66	**80**	**347**		

C.3 Checking for TCC Constraints

The optimized candidate solutions are only good for the generation/load balance criteria. For safety, we need to check the TCC constraints to make sure the solution is actually realizable. Usually, there will be some good solution in the candidate solutions. If one of the top solutions does not work, we just need to check the rest and usually we can find a good one. In the worst case, if all of the candidate solutions do not satisfy the TCC constraint, we can save all of these candidate solutions to the bad solutions table and run the program again. Since the algorithm can avoid solutions in the bad solution table, we can find some other candidate solutions for TCC check. Since the regular power flow calculation may take too much time for large scale power systems, it is necessary to apply some approximate algorithms to get timely results.

TABLE IV
TOP 10 SOLUTIONS TO SPLIT THE SYSTEM INTO THREE ISLANDS

No	Islands Info		Opened Lines
	Buses	P_G/P_L (MW)	
1	59 (1~20, 26, 30~31, 33~67, 117)	1823/1791	17-113, 20-21, 25-26, 29-31, 31-32, 47-69, 49-69, 65-68
	59 (21~25, 27~29, 32, 68~116, 118)	1962.4/1841	
2	39 (1~32, 34, 36, 72, 113~115, 117)	1022/1006	15-33, 24-70, 30-38, 34-37, 34-43, 35-36, 71-72
	79 (33, 35, 37~71, 73~112, 116, 118)	2763.4/2626	
3	80 (1~75, 113~117)	2559.4/2425	68-81, 69-77, 75-77, 76-118
	38 (76~112, 118)	1226/1207	
4	34 (1~21, 25~33, 113~115, 117)	1022/897	1934, 21-22, 23-25, 23-32, 30-38, 33-37
	84 (22~24, 34~112, 116, 118)	2763.4/2735	
5	90 (1~83, 99, 113~118)	2906.4/2813	80-96 80-97, 80-98, 82-96, 83-84, 83-85, 99-100
	28 (84~98, 100~112)	879/819	
6	95 (1~87, 96~97, 113~118)	2910.4/2880	80-98, 80-99, 85-88, 85-89, 94-96, 95-96
	23 (88~95, 98~112)	875/752	
7	17 (1~16, 117)	488/483	8-30, 15-17, 15-19, 15-33, 16-17
	101 (17~116, 118)	3297.4/3149	
8	101 (1~22, 30, 33~69, 75~112, 116~118)	3251.4/3217	17-31, 17-113, 22-23, 26-30, 69-70, 70-75, 74-75
	17 (23~29, 31~32, 70~74, 113~115)	534/415	
9	12 (3~11, 13~15)	450/367	1-3, 3-12, 7-12, 8-30, 11-12, 12-14, 15-17, 15-19, 15-33
	106 (1~2, 12, 16~118)	3335.4/3265	
10	113 (1~24, 26, 30~114, 116~118)	3565.4/3498	23-25, 25-26, 27-32, 29-31, 114-115
	5 (25, 27~29, 115)	220/134	

V. Conclusion

This paper proposes a binary particle swarm optimization based power system splitting algorithm. Simulation studies demonstrate that the algorithm is computationally efficient and is computationally efficient and is able to find a satisfactory solution, thus feasible for real time implementation. Given a suitable measure for dynamic response behavior, the algorithm proposed in this paper will be able to consider dynamic responses during the islanding operation.

References

[1] J. Jung, C. C. Liu, S. L. Tanimoto, and V. Vittal, "Adaptation in load shedding under vulnerable operating conditions," IEEE Transactions on Power Systems, vol. 17, no. 4, pp. 1199-1205, November 2002.

[2] H. You, V. Vittal, and Z. Yang, "Self-healing in power systems: an approach using islanding and rate of frequency decline-based load shedding," IEEE Transactions on Power Systems, vol. 18, no. 1, pp. 174-181, February 2003.

[3] H. You, V. Vittal, and X. Wang, "Slow coherency-based islanding," IEEE Transactions on Power Systems, vol. 19, no. 1, pp. 183-491, February 2004.

[4] X. Wang, and V. Vittal, "System islanding using minimal cutsets with minimum net flow," 2004 IEEE PES Power Systems Conference and Exposition, vol. 1, pp. 379-384, October 2004.

[5] Q. Zhao, K. Sun, D. Zheng, J. Ma, and Q. Lu, "A study of system splitting strategies for island operation of power system: a two-phase method based on OBDDs," IEEE Transactions on Power Systems, vol. 18, no. 4, pp. 1556, November 2003.

[6] K. Sun, D. Zheng, and Q. Lu, "Splitting strategies for islanding operation of large-scale power systems using OBDD-based methods," IEEE Transactions on Power Systems, vol. 18, No.2, pp. 912-923, May 2003.

[7] ieee30cdf.txt [Online]. Available: http://www.ee.washington.edu/re-search /pstca/.

[8] ieee118cdf.txt [Online]. Available: http://www.ee.washington.edu/re-search /pstca/.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," IEEE International Conference on Neural Networks, vol. IV, pp. 1942-1948, Perth, Australia, 1995.

[10] Zwe-Lee Gaing, "Discrete particle swarm optimization algorithm for unit commitment", 2003 IEEE Power Engineering Society General Meeting, vol. 1, pp. 13-17, Toronto, Canada, July 2003.

[11] J. Kennedy and R. Eberhart, " A Discrete Binary Version of the Particle Swarm Optimization," Proceedings of IEEE International Conference on Neural Networks, vol. IV, pp. 4104-4108, Perth, Australia, 1997.