01 Apr 2009

Real-Time Implementation of an Intelligent Algorithm for Electric Ship Power System Reconfiguration

Pinaki Mitra

Ganesh K. Venayagamoorthy
Missouri University of Science and Technology

Recommended Citation

P. Mitra and G. K. Venayagamoorthy, "Real-Time Implementation of an Intelligent Algorithm for Electric Ship Power System Reconfiguration," *Proceedings of the IEEE Electric Ship Technologies Symposium, 2009. ESTS 2009*, Institute of Electrical and Electronics Engineers (IEEE), Apr 2009.
The definitive version is available at https://doi.org/10.1109/ESTS.2009.4906519

Real-Time Implementation of an Intelligent Algorithm for Electric Ship Power System Reconfiguration

Pinaki Mitra and Ganesh K. Venayagamoorthy
Real-Time Power and Intelligent Systems Laboratory,
Department of Electrical and Computer Engineering,
Missouri University of Science and Technology,
Rolla, Missouri, USA.
pm33d@mst.edu, gkumar@ieee.org

***Abstract*— The naval electric ship is often subject to severe damages under battle conditions. The damages or faults might even affect the generators and as a result, critical loads might suffer from power deficiency which may lead to an eventual collapse of rest of the system. In order to serve the critical loads and maintain a proper power balance without excessive generation, the ship power system requires a fast reconfiguration of the remaining system under fault conditions. A fast intelligent algorithm using the Small Population based Particle Swarm Optimization (SPPSO) for dynamic reconfiguration of the available generators and loads when a fault in the ship power system is detected is presented in this paper. SPPSO is a variant of PSO which operates with fewer particles and a regeneration concept, where new potential solutions are generated every few iterations. This concept of regeneration makes the algorithm fast and enhances its exploration capability to a large extent. The strength of the proposed reconfiguration strategy is first illustrated with Matlab results and then with a real-time implementation on a real time digital simulator and a digital signal processor.**

I. INTRODUCTION

Reconfiguration of distribution power network is a well-known research area in power system. Conventionally it is viewed as a multi-objective optimization problem [1]. The classical approach to solve this distribution system reconfiguration problem is using heuristic methods [2]-[3]. Due to the stochastic nature of the problem, computational intelligence algorithms, such as genetic algorithm (GA), particle swarm optimization (PSO), differential evolution, ant colony optimization, a hybrid of artificial immune system and ant colony optimization have been used by different researchers as in the references [1] and [4]-[8]. But, there are few basic differences between normal distribution system and a naval shipboard power system. In navy ships, there are several emergency loads which must be powered during battle conditions. Also, the reconfiguration of the ship power system should be very fast so that the quality of power to those critical loads is maintained at desired level all the time. These particularities of a ship power system necessitate a simple, fast and intelligent reconfiguration strategy, which can be easily implemented in real-time to produce desired result. Many researchers are presently working in the area of dynamic reconfiguration of the ship power system. In [10], a fast reconfiguration algorithm is proposed which is based on zone-based differential protection system. This algorithm has two consecutive search functions. The first one is a path search algorithm and the second one is a load shedding scheme based on load priorities for the path having negative power balance. But, no investigation in a real-time platform is reported. Therefore, it is difficult to predict how much time the algorithm would require to change the status of the breakers in a real system. This work is further developed in [11] and [12] by applying binary PSO and GA, respectively, for the load shedding scheme proposed in [10]. Generally, both PSO and GA are based on a number of candidate solutions ('chromosomes' in GA and 'particles' in PSO). The exploration becomes better, if the number of potential solutions is increased. But this eventually makes the algorithm slow and it is not practical for the real-time applications. Recently, agent based reconfiguration strategies have been proposed in [13] and [14].

A less complex approach for reconfiguration which is fast enough to implement in real time without serious deterioration in power quality is presented in this paper. The approach is based on Small Population based Particle Swarm Optimization (SPPSO). The proposed approach is first validated through Matlab based study. Thereafter, a real-time implementation is carried out on a Real Time Digital Simulator (RTDS) and Digital Signal Processor (DSP) based platform.

The rest of the paper is organized as follows: The proposed intelligent reconfiguration algorithm is discussed in the section II. The working principle of SPPSO is discussed in section III. The test system and typical results are presented in section IV. Finally, the conclusions and future work are given in section V.

This work was supported by US Office of Naval Research under the Young Investigator Program award # N00014-07-1-0806

II. INTELLIGENT RECONFIGURATION ALGORITHM

Ship power system consists of two main generators of 36 MW (MTG1 and MTG2), two auxiliary generators of 4 MW (ATG1 and ATG2) and several critical and non-critical loads. A typical structure is shown in Fig. 1, where the loads are represented as lumped loads at eight buses of the network. This representation has 20 circuit breakers among which four are generator breakers, eight are load breakers and remaining eight are path breakers. The status of the breakers can be either 'CLOSED' or 'OPEN' and hence theoretically there are 2^{20} possibilities for the breaker positions. The breaker positions must also satisfy the following condition

$$P_{GEN} \geq P_{LOAD} \tag{1}$$

where P_{GEN} is the available generation at a particular time and P_{LOAD} is the amount of load to be powered at that point of time which is referred as 'available load' in the rest of the paper. When a fault occurs at a bus, the protection system senses the fault and trips the breakers associated with the fault to isolate it. The available breaker status is thus modified with the fault. The available generation and load profile of the system also changes simultaneously. Based on these changes, the reconfiguration strategy now searches for a new topology of the ship power system, so that it can supply maximum number of the critical loads and also with optimal generation. The objective functions for this problem are formulated as follows:

$$Max\left(\sum_{i=1}^{N} p_i \cdot L_i\right) \tag{2}$$

$$\text{and} \quad Min\left(P_{GEN} - P_{LOAD}\right) \tag{3}$$

where p_i is the priority weighting associated with a load L_i and N is the total number of loads. A lower priority weighting signifies a lower priority. The proposed reconfiguration strategy consists of following steps:

Step 1: First, the configuration of the ship power system is represented by a 1 x 20 matrix consisting of only binary digits, where '1' represents the 'CLOSED' and '0' represents the 'OPEN' status of the breakers respectively. After getting the fault information, it updates the matrix accordingly.

Step 2: Now, three distinct matrices are produced from the original one – one representing generator breaker status, another representing the load breaker status and the last one representing the bus connection breaker status respectively. This is carried out to reduce the complexity of the search space for the reconfiguration algorithm.

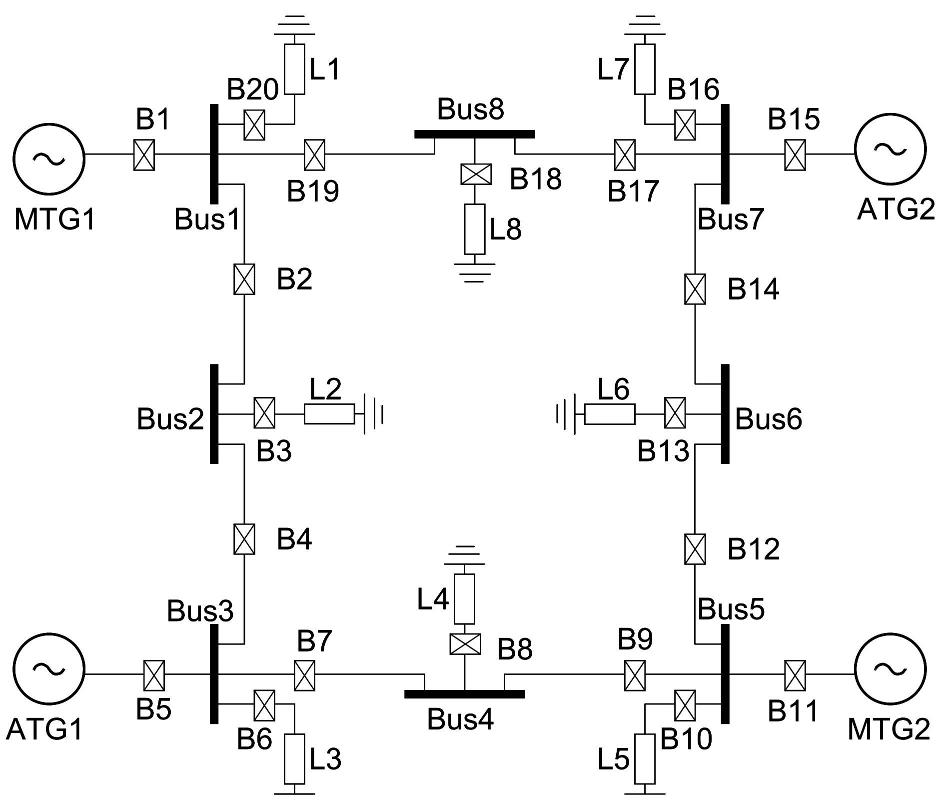

Figure 1. Structure of ship power system having eight buses (Bus1 to Bus8), four generators (MTG1, MTG2, ATG1 and ATG2), twenty breakers (B1 to B20), and eight loads (L1 to L8)

Step 3: From the updated generator and load breaker matrices, the total available generation and the total available load are calculated.

Step 4: If $P_{GEN} \geq P_{LOAD,}$ all the load breakers (except those tripped by the fault) are closed. If $P_{GEN} < P_{LOAD}$, all the generator breakers are to be closed (except the faulted generator(s), if any).

Step 5: The above step 4 further reduces the search space complexity. For $P_{GEN} \geq P_{LOAD}$, the proposed strategy searches for the optimum generation (guided by the objective function in (3)) within a very small search space of 2^M options, where M = 4 in Fig. 1. Hence, for this purpose no intelligent algorithm is needed. For $P_{GEN} < P_{LOAD}$, the proposed strategy carried out an optimal load shedding using the objective function in (2). The search space for this becomes 2^N, where N = 8 in Fig. 1 But number of loads can be more in a real system and with addition of one load, the search space becomes double. Therefore, in order to provide a generalized solution, intelligent techniques capable of making fast decisions are preferred. In this paper, this load reconfiguration is carried out by SPPSO algorithm [15] for maximizing (2).

III. Small Population Based PSO

A. Conventional Particle Swarm Optimization Algorithm

Particle swarm optimization is a population based search algorithm which aims to replicate the motion of flock of birds and school of fishes [16]. A swarm is considered to be a collection of particles, where each particle represents a potential solution to the problem. The particle changes its position within the swarm based on the experience and knowledge of its neighbors. Basically it 'flies' over the search space to find the optimal solution [16].

Initially a population of random solutions is considered. A random velocity is also assigned to each individual particle with which they start flying within the search space. Also, each particle has a memory which keeps track of the previous best position of the particle and the corresponding fitness. This previous best value is called 'p_{best}'. There is another value called 'g_{best}', which is the best value of all the 'p_{best}' values of the particles in the swarm. The fundamental concept of the PSO technique is that the particles always accelerate towards their 'p_{best}' and 'g_{best}' positions at each time step. Fig. 2 demonstrates the concept of PSO where,

a) $x_{id}(k)$ is the current position of i^{th} particle with d dimensions at instant k.
b) $x_{id}(k+1)$ is the position of i^{th} particle with d dimensions at instant $(k+1)$.
c) $v_{id}(k)$ is the initial velocity of the i^{th} particle with d dimensions at instant k.
d) $v_{id}(k+1)$ is the initial velocity of the i^{th} particle with d dimensions at instant $(k+1)$.
e) w is the inertia weight which stands for the tendency of the particle to maintain its previous position.
f) c_1 is the cognitive acceleration constant, which stands for the particles' tendency to move towards its 'p_{best}' position.

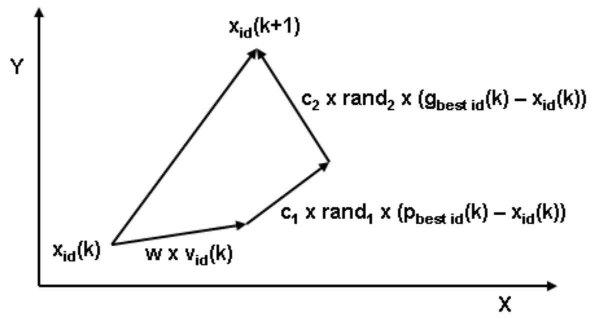

Figure 2. Concept of changing a particle's position in two dimensions

g) c_2 is the social acceleration constant which represents the tendency of the particle to move towards the 'g_{best}' position.

The velocity and the position of the particle are updated according to the following equations. The velocity of the i^{th} particle of d dimension is given by:

$$v_{id}(k+1) = w \cdot v_{id}(k) + c_1 \cdot rand_1 \cdot (p_{best\_id}(k) - x_{id}(k)) + c_2 \cdot rand_2 \cdot (g_{best\_id}(k) - x_{id}(k)) \quad (4)$$

The position vector of the i^{th} particle of d dimension is updated as follows:

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1) \quad (5)$$

B. Small Population Based PSO

As the number of particles in the swarm increases, the convergence to a global solution is more and more ensured. The reason is, higher the number of particles, the greater the exploration of the search space. But, as the number of particles increases, the memory requirement for the algorithm also increases which is often not permissible in the real world application of the algorithm with digital signal processors or microcontrollers, etc. Also, the speed of convergence reduces a lot. In order to overcome these problems, SPPSO algorithm was developed by Das and Venayagamoorthy in [15]. The concept of SPPSO is to start with a small number of particles (generally around five) and after a few iterations, replace all the particles except the global best with same number of regenerated particles. In this method, since the PSO runs with a very small number of particles, the memory requirement is reduced a lot. Also, since after few iterations a new set of particles are introduced, the chance of fixation to a local minima decreases and convergence is achieved much faster than conventional PSO.

IV. Test System and Results

The performance of the proposed reconfiguration strategy is demonstrated on two research environments – Matlab and RTDS. In both cases, the test system is similar to that represented by the single line diagram in Fig. 1. The only difference is, in case of Matlab based study, a system with six loads is tested first and then a system with eight loads is considered. Whereas, in case of RTDS based study, only the system with eight loads is considered. For the system with six loads, L4 and L8 of Fig. 1 are removed. The power system setup for this case is shown in Fig. 3.

A. Matlab Based Case Study:

For the test system presented in Fig. 3 (having six loads), two different combinations of load magnitudes and load priorities are considered. Those are referred as cases 1 and 2, and are presented in Tables I and II respectively. For each case, fault is created arbitrarily at different buses. The breaker status is accordingly changed. The post-fault breaker status is then sent to the reconfiguration algorithm. The reconfiguration algorithm updates the breaker status matrix. For the sake of convenience, it is assumed that all the breakers were in 'CLOSED' state before the creation of the fault. Tables III and IV correspond to the fault scenarios and outputs from the reconfiguration algorithm for cases 1 and 2 respectively.

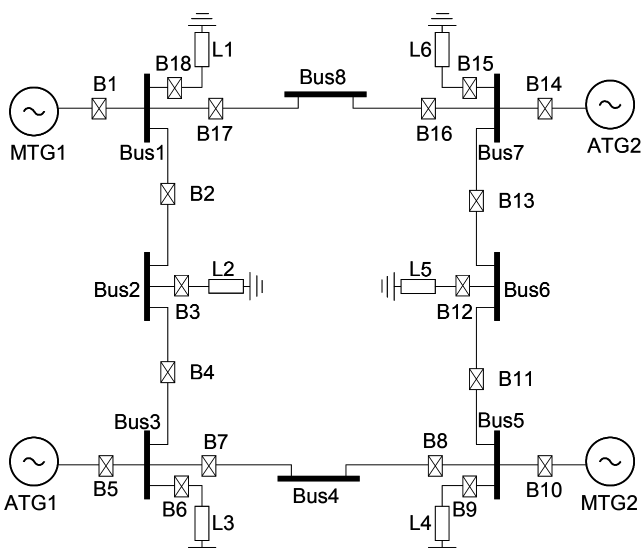

Figure 3. Test system with six loads

TABLE I. LOAD MAGNITUDE AND PRIORITIES FOR CASE 1

Load No.	L1	L2	L3	L4	L5	L6
Magnitude (MW)	2	20	2	2	20	2
Priority Weighting	1	2	1	2	2	2

TABLE II. LOAD MAGNITUDE AND PRIORITIES FOR CASE 2

Load No.	L1	L2	L3	L4	L5	L6
Magnitude (MW)	2	20	5	2	20	4
Priority Weighting	1	2	2	3	2	1

In fault scenario 1, a fault is created at Bus 1 (in Fig. 3). Thus, the generator MTG1 of 36 MW and load L1 of 2 MW are tripped. Now in Fig. 3, the total available generation is 44 MW and the total available load is 46 MW. This definitely requires a load of at least 2 MW to be shed. For the sake of simplicity, generation reserve is not considered. Now, looking at the load priority weightings in Table I, it is clear that L3 has the least priority to be powered. The reconfiguration algorithm also recommends the shedding of L3. To verify that the reconfiguration algorithm is consistent in recommending the same load to be shed every time, 50 trials are carried out. It is found that each time it is generating the same solution. The average reconfiguration time is given in the Table III. The studies are carried out on a PC with an Intel Pentium IV 2.80 GHz processor. The variation of the inverse of the cost function (in (2)) with number of fitness evaluation is shown in Fig. 4. In fault scenario 2, a fault at Bus 2 is applied. But no generator is associated with the bus. For this fault, only load L2 of 20 MW is tripped. Since the total available generation is 80 MW and total available load is only 28 MW, the reconfiguration algorithm recommends the tripping of all the generators except MTG1 of 36 MW capacity since this is sufficient to serve the total available load.

Now, a fault at Bus 1 is applied for case 2. In case 2, with the application of fault, the available load becomes 51 MW and the available generation is 44 MW. This calls for tripping of at least 7 MW of load (without considering any reserve). From the load magnitude table (Table II), it appears that the tripping of the load L3 of 5 MW and the load L4 of 2 MW will solve this problem. But, if the load priority weighting is considered, the priority weighting of load L6 is lower than that of L4, and the product of L4 and its priority weighting is greater than the product of L6 and its priority weighting. Hence the algorithm recommends tripping loads L3 and L6 instead of loads L3 and L4. The inverse of the cost function with the number of fitness evaluation is shown in Fig. 5.

TABLE III. THE OUTPUT OF RECONFIGURATION ALGORITHM FOR CASE 1

Faulted Bus	Total Available Generation (MW)	Total Available Load (MW)	Possible Generator Breaker Matrix	Possible Load Breaker Matrix	Suggestion For Load-shedding	Average Reconfiguration Time (ms)
1	44	46	0111	010111	L3	70.3
2	80	28	1000	101111	None	11.9

TABLE IV. THE OUTPUT OF RECONFIGURATION ALGORITHM FOR CASE 2

Faulted Bus	Total Available Generation (MW)	Total Available Load (MW)	Possible Generator Breaker Matrix	Possible Load Breaker Matrix	Suggestion For Load-shedding	Average Computation Time (ms)
1	44	51	0111	010110	L3, L6	75.5

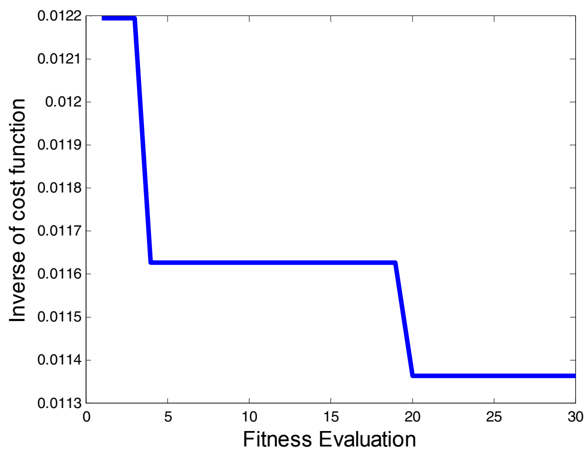

Figure 4. Inverse of cost function vs. iteration curve for case 1

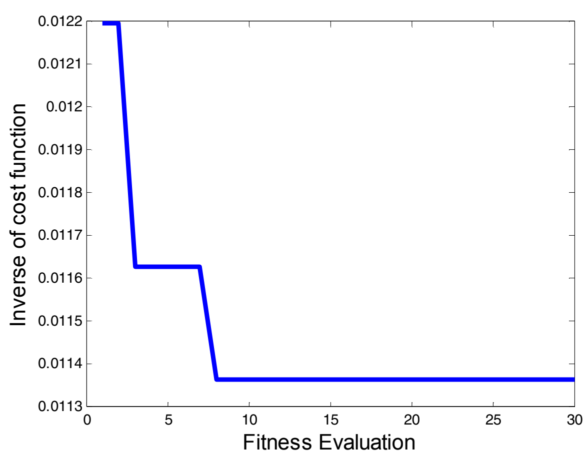

Figure 5. Inverse of cost function vs. iteration curve for case 2

To make the situation a little more complex, a system with eight loads (Fig. 1) is now considered. The load magnitude and priorities are chosen arbitrarily and are shown in Table V and this case is referred as case 3. Here the search space for the SPPSO algorithm becomes 2^8. With the application of fault at Bus 1 again, the available load becomes 66 MW and the available generation is 44 MW. This requires tripping of 22 MW of load. The reconfiguration algorithm correctly recommends two possible solutions. These are tripping of either loads L3 and L6 or L6 and L7, in order to maximize the cost function in (2). Fig. 6 shows the variation of the inverse of cost function with number of fitness evaluations for case 3 and Table VI shows the output of the algorithm.

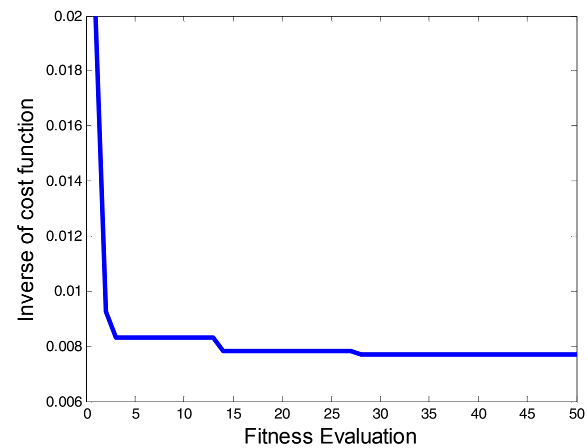

Figure 6. Inverse of cost function vs. iteration curve for case 3

TABLE V. LOAD MAGNITUDE AND PRIORITIES FOR CASE 3

Load No.	L1	L2	L3	L4	L5	L6	L7	L8
Magnitude (MW)	2	20	2	10	2	20	2	10
Priority Weighting	6	4	4	1	6	1	4	2

TABLE VI. THE OUTPUT OF RECONFIGURATION ALGORITHM FOR CASE 3

Faulted Bus	Total Available Generation (MW)	Total Available Load (MW)	Possible Generator Breaker Matrix	Possible Load Breaker Matrix	Suggestion For Load-shedding	Average Computation Time (ms)
1	44	66	0111	01011011 or 01111001	L3, L6 or L6, L7	119.1

In all the above cases, the regeneration concept in SPPSO algorithm is carried out every other five iterations. The total number of iterations for cases 1 and 2 is 30, and for case 3, it is 50. Even with a smaller number of iterations, the success rate of the algorithm (recommending feasible solutions) is found to be 100%. With the increase in the size of the power system, the iterations required to attain a 100% success rate is foreseen to increase. But there is no doubt that this algorithm will still remain fast enough to find out the global solution within tolerable limit.

B. RTDS Based Case Study:

The model of an electric ship power system shown in Fig. 1 is built on the RTDS environment. The advantage of the RTDS is that, it can represent the dynamics of a power system almost as close as that of a practical system. The real time experimental setup is shown in Fig. 7. The breaker status signals from the RTDS are sent to the DSP. Using these signals, the reconfiguration algorithm implemented on the DSP recommends new breaker status, if necessary.

The same fault at Bus 1 as in the Matlab study is now applied from the RSCAD (a RTDS module) runtime window. The same results as observed in the Matlab study are obtained. Since the computation of the algorithm was very fast, the system had to run under overloaded condition for a very small period of time. Thus, there is no significant deterioration in the active power and voltage profile of the high priority loads. In order to demonstrate the impact of reconfiguration on critical loads, as well as on the entire system, a high priority load L2 is selected. Before the occurrence of a fault, L2 was consuming 19 MW of power at a voltage of 0.98 p.u. Post-fault, there was a transient in the power consumed and voltage of L2 but it settled within two seconds. The dynamic variation of power and voltage at L2 load bus is shown in Figs. 8 and 9 respectively. Those are compared with the case where no reconfiguration is carried out. It is observed that the system becomes unstable if no reconfiguration is carried out. Also, the impact on speed of MTG2 is observed and compared to the case without reconfiguration (Fig. 10) and it is found that the speed oscillation is also not significant due to the fast reconfiguration by SPPSO algorithm.

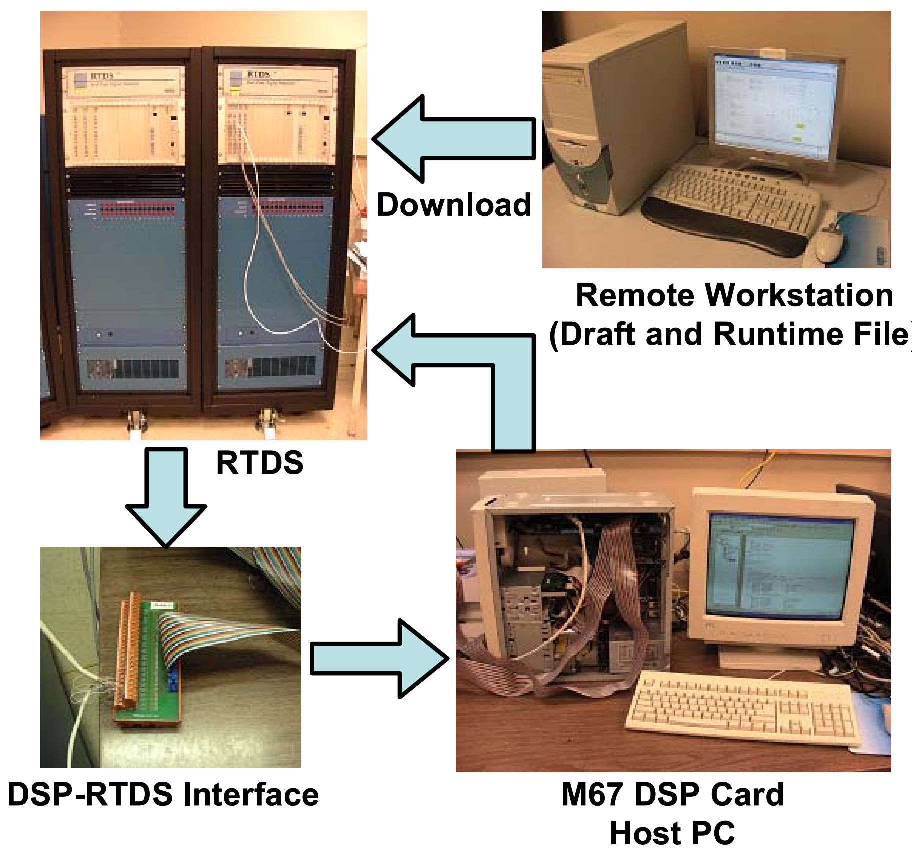

Figure 7. Laboratory experimental setup

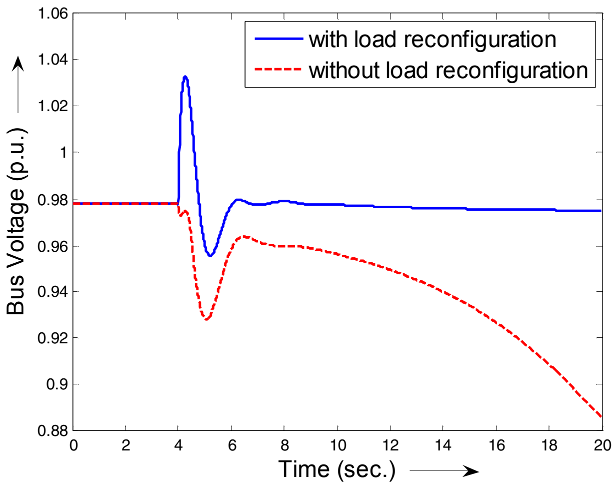

Figure 8. Bus 2 voltage characteristics of load L2 post-fault

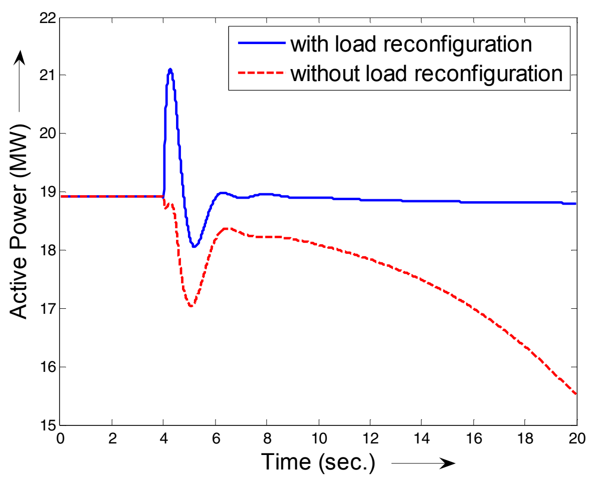

Figure 9. Active power characteristics of load L2 post-fault

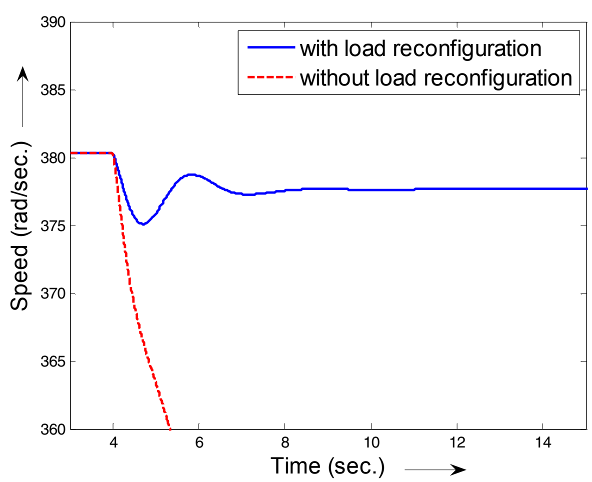

Figure 10. Speed characteristics of MTG1 post-fault

V. Conclusion and Scope of Future Work

An intelligent dynamic generator and load reconfiguration strategy for an electric ship power system has been presented in this paper. The dynamic reconfiguration is carried out using the small population based particle swarm optimization. The presented strategy is simple, fast and easy to implement for real-time applications. The speed of reconfiguration strategy is enhanced using few simple logical steps that reduce the search space complexity to a large extent. Studies in Matlab and real-time environment are performed to illustrate the capability of the proposed reconfiguration strategy. Different fault conditions and load priority combinations have demonstrated that the algorithm is robust in determining optimal solutions. Furthermore, with the reconfiguration, the effects of faults in system on the bus voltages and power flows are minimized.

In future, more complex cases, like the occurrence of multiple faults simultaneously on different buses, which may result in two or more islanded systems, are to be studied. This would require a path search algorithm to be included in the reconfiguration strategy. Maximizing the amount of load and maximizing the number of critical loads are sometimes conflicting. Therefore, the concept of Pareto optimality can be applied to solve such conflicting objectives.

References

[1] A. Ahuja, S. Das, and A. Pahwa, “An AIS-ACO Hybrid Approach for Multi-Objective Distribution System Reconfiguration”, *IEEE Trans. Power Syst.*, vol. 22, no. 3, August 2007, pp. 1101-1111.

[2] S. K. Goswami and S. K. Basu, “A new algorithm for the reconfiguration of distribution feeders for loss minimization,” *IEEE Trans. Power Del.*, vol. 7, no. 3, pp. 1484–1491, Jul. 1992.

[3] F. V. Gomes and S. Carneiro, Jr., “A new heuristic reconfiguration algorithm for large distribution systems,” *IEEE Trans. Power Syst.*, vol. 20, no. 3, pp. 1373–1378, Aug. 2005.

[4] J. H. Choi and J. C. Kim, “Network reconfiguration at the power system with dispersed generations for loss reduction,” in *Proc. IEEE Power Eng. Soc. Meeting*, Jan. 23–27, 2000, vol. 4, pp. 2363–2367.

[5] N. Kagan and C. C. B. de Oliveira, “Fuzzy decision model for the reconfiguration of distribution networks using genetic algorithms,” in *Proc. 13th Power System Computation Conf.*, Trondheim, Norway, 1999.

[6] X. Jin, J. Zhao, Y. Sun, K. Li, and B. Zhang, “Distribution network reconfiguration for load balancing using binary particle swarm optimization,” in *Proc. Int. Conf. Power System Technology*, Nov. 2004.

[7] J. P. Chiou, C.-F. Chang, and C.-T. S, “Variable scaling hybrid differential evolution for solving network reconfiguration of distribution systems,” *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 668–674, May 2005.

[8] A. Ahuja and A. Pahwa, “Using ant colony optimization for loss minimization in distribution networks,” in *Proc. 37th North Amer. Power Symp.*, Oct. 2005, pp. 470–474.

[9] E. Carpaneto and G. Chicco, “Distribution System Minimum Loss Reconfiguration in the Hyper-Cube Ant Colony Optimization Framework”, *Electric Power System Research* 78 (2008), pp. 2037-2045.

[10] Yan Haung, “Fast Reconfiguration Algorithm Development for Shipboard Power Systems”, *Master of Science Thesis*, Mississippi State University, 2006.

[11] N. Kumar, A. and N. N. Schulz, “Shipboard Power System Restoration Using Binary Particle Swarm Optimization”, *Proceedings of the 39th North American Power Symposium, NAPS ’07*, Sept 30-Oct 2, 2007, pp. 164-169.

[12] K. R. Padamati, N. N. Schulz and A. K. Srivastava, "Application of Genetic Algorithm for Reconfiguration of Shipboard Power System", *Proceedings of the 39th North American Power Symposium, NAPS '07*, Sept 30-Oct 2, 2007, pp. 159-163.

[13] J. M. Solanki and N. N. Schulz, "Using Intelligent Multi-Agent Systems for Shipboard Power Systems Reconfiguration", ISAP 2005, pp. 212-214.

[14] Y. Qudaih and T. Hiyama, "Reconfiguration of Power Distribution System Using Multi Agent and Hierarchical Based Load Following Operation with Energy Capacitor System", *The 8th International Power Engineering Conference,* IPEC 2007, pp. 223-227.

[15] T. K. Das, G. K. Venayagamoorthy, U. O. Aliyu, "Bio-Inspired Algorithms for the Design of Multiple Optimal Power System Stabilizers: SPPSO and BFA", *IEEE Transactions on Industry Applications*, Vol. 44, Issue 5, Sept-Oct 2008, pp. 1445-1457.

[16] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, R. G. Harley, "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems", *IEEE Transactions on Evolutionary Computation,* available on *IEEExplore*.