Implementation of an Intelligent Reconfiguration Algorithm for an Electric Ship Power System

Pinaki Mitra
Student Member, IEEE
Real-Time Power and Intelligent Systems Laboratory
Missouri University of Science and Technology
Rolla, MO 65401, USA
pinakimitra@ieee.org

Ganesh K. Venayagamoorthy
Senior Member, IEEE
Real-Time Power and Intelligent Systems Laboratory
Missouri University of Science and Technology
Rolla, MO 65401, USA
gkumar@ ieee.org

***Abstract* -- In all-electric navy ships, severe damages or faults may occur during battle conditions. This might even affect the generators and as a result, critical loads might suffer from power deficiency for a long time and ultimately lead to a complete system collapse. A fast reconfiguration of the power path is therefore necessary in order to serve the critical loads and to maintain a proper power balance in ship power system. This paper proposes a fast, intelligent reconfiguration algorithm, where Pareto optimal solutions are obtained by Small Population based Particle Swarm Optimization (SPPSO) from two conflicting objective functions. From the Pareto set, the final solution is chosen depending on users' preference. SPPSO is a variant of PSO which works with very few numbers of particles with a regeneration of new solutions within the search space after few iterations. This concept of regeneration in SPPSO make the algorithm really fast and enhances its capability to a large extent. The strength of the proposed reconfiguration strategy is tested in Real-Time Digital Simulator (RTDS) environment.**

***Index Terms*-- Dynamic reconfiguration, Electric ship power system, Parento optimal solutions, Small population based particle swarm optimization.**

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 through heuristic search algorithms [2]-[3]. Due to the stochastic nature of the problem, computational intelligence algorithms, such as genetic algorithm, particle swarm optimization, differential evolution, ant colony optimization, a hybrid of artificial immune system and ant colony optimization have been used by different researchers in [1] and [4]-[9] respectively. 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 served 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, the paper does not present any real-time study. So, it is tough to predict how much time the algorithm will take to change the status of the breakers in real system. This work is further developed in [11] and [12] by applying binary PSO and genetic algorithm respectively for the load shedding scheme proposed in [10]. Generally, both PSO and GA work with 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 becomes unfit for the real-time applications. Other than these, agent based reconfiguration strategies have been proposed in [13] and [14].

This paper proposes a new simpler approach for reconfiguration which is fast enough to implement in real time without serious deterioration in power quality. To enhance the speed of execution of the algorithm in a multi-objective framework, a set of Pareto optimal solutions are first extracted by Small Population based Particle Swarm Optimization (SPPSO) from two conflicting objectives. Those Pareto optimal solutions present a set of permissible operating modes. After that, those solutions are passed through a set of questions representing user's preference regarding the operating modes. Based on the response to those questions, the final solution is obtained. The concept of including users' preference in solving multi-objective optimization problem was introduced by Tanaka et. al. [15]. That was an offline process involving human interaction. But, in ship system, the speed of reconfiguration is the most important factor. So, human participation is not possible and full automation is required. Hence, in this paper, the concept of users' preference is included inside the algorithm itself in the form of knowledge base. A real-time implementation of the algorithm on a Real Time Digital Simulator (RTDS) and a DSP platform is presented.

The paper is organized as follows: Section II discusses the detail of the proposed algorithm. Section III describes the working principle of Small Population Based Particle Swarm

Optimization. Section IV presents the test system and typical results. Finally the conclusion and future work are summarized in section V.

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: a) maximum number of the critical loads, b) maximum amount of load and c) also with optimal generation. The objective functions for this problem are formulated as follows:

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

$$Max\left(\sum_{i=1}^{N} L_i\right) \tag{3}$$

$$\text{and} \quad Min\left(P_{GEN} - P_{LOAD}\right) \tag{4}$$

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.

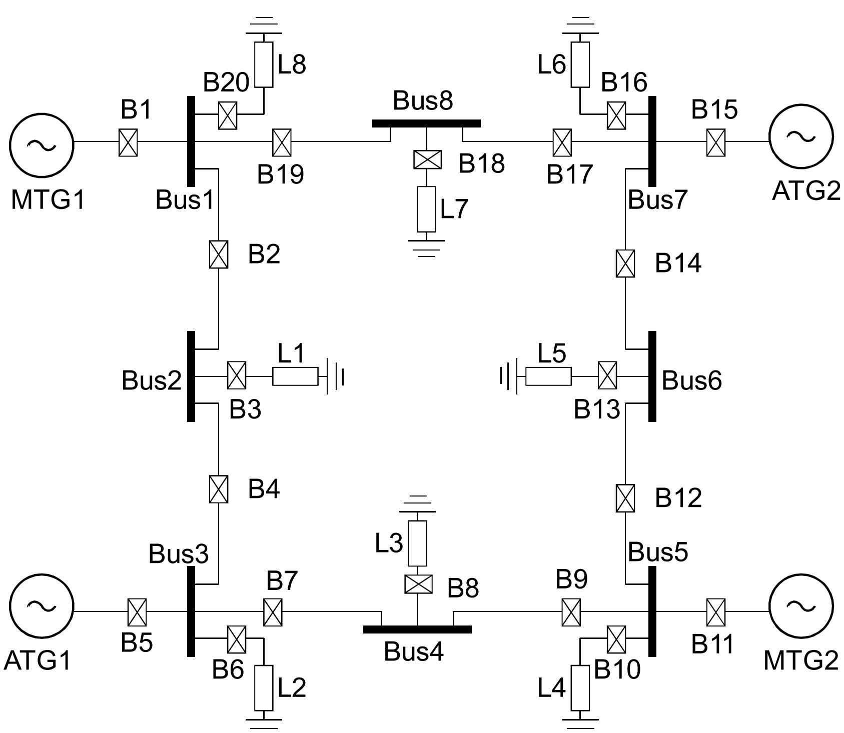

Fig. 1. Structure of ship power system having eight buses (Bus1 to Bus8), four generators (MTG1, MTG2, ATG1 and ATG2), twenty breakers (B1 to B20), and eight loads (L1 to L8)

In practice objectives (2) and (3) may be contradictory in many cases. For example, if there is a high priority load of 2 MW and a low priority load of 20 MW, during load shedding, there are clearly two conflicting choices. If the load priority is important, then one will go for shedding the 20 MW load. But this will not be a preferred choice in terms of the second objective. In order to resolve these types of multi-objective optimization problems, one has to find out the Pareto optimal front. The concept of Pareto optimality is as follows:

In a general multiobjective optimization problem, the objectives are to be achieved simultaneously and they are formulated as [16]

$$\underset{x}{Minimize}\, f_i(x) \qquad i = 1,....,N_{obj} \tag{5}$$

$$Subject\ to : \begin{cases} g_j(x) = 0 & j = 1,....,M \\ h_k(x) \leq 0 & k = 1,....,K \end{cases} \tag{6}$$

Where f_i is the i^{th} objective function, x is a decision vector that represents a solution; N_{obj} is the number of objectives. M and K are the numbers of equality and inequality constraints, respectively. Now, any two solutions x_1 and x_2 of a multiobjective optimization problem can have one of two possibilities: one dominates the other or none dominates the other. Without losing generality, in a minimization problem, a solution x_1 dominates x_2 if the following two conditions are satisfied:

$$\forall i \in \{1,2,...,N_{obj}\} : f_i(x_1) \leq f_i(x_2) \tag{7}$$

$$\exists j \in \{1,2,...,N_{obj}\} : f_j(x_1) < f_j(x_2) \tag{8}$$

If any of the above condition is violated, x_1 does not dominate x_2. If x_1 dominates x_2, x_1 is called the non-dominated solution. A set of all non-dominated solution inside the search space is called Pareto optimal set or Pareto optimal front.

There are several methods of extraction of Pareto optimal front in a multiobjective optimization problem [17]-[19]. But in most of the techniques, the process of finding non-dominated solution is a computationally complex and time consuming process. The easiest method for a two-objective problem is to represent the weighted sum of two objectives as follows [20]:

$$f(x) = w_1 f_1(x) + w_2 f_2(x) \tag{9}$$

Where,

$$w_1 + w_2 = 1 \tag{10}$$

In this way, the two objectives are expressed as a single objective which can be optimized with any conventional or evolutionary methods. If w_1 is now varied from 0 to 1 in small steps and for each value of w_1 the optimum value of $f(x)$ is calculated, that gives us the entire set of Pareto optimal solutions.

Generally, the optimization algorithms are used offline and hence time is not a constraint for those applications. But, in this paper, the reconfiguration algorithm, which includes the extraction of Pareto front and then taking the best decision, is to be performed online within a very small time. Otherwise, the critical loads will suffer from power deficiencies, which is not acceptable in battle situations. In order to enhance the speed of the reconfiguration process, few measures are taken. The proposed reconfiguration strategy goes through 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.

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 (4)) 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 carries out an optimal load shedding using the objective functions in (2) and (3). As discussed earlier, these objectives are sometimes conflicting and hence extraction of Pareto front is necessary. The search space for this Pareto front extraction 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 Pareto optimal front extraction is carried out by SPPSO algorithm [21].

Step 6: The problem discussed in this paper is a discrete optimization problem. Hence, the Pareto front is actually a set of discrete solutions which indicates the permissible operating modes of the ship power system. Now, those solutions are passed through a set of predefined questions which represent the user's preference regarding the operating mode. Based on the response to those questions, the final solution is selected. For example, in battle mode, user may need a high value of priority loads (such as weapon loads, radar loads, etc.) and a critical minimum value of total amount of load to be served. The Pareto solution which satisfies this condition is now selected. Conversely, in normal mode, user may need a critical minimum value of priority loads (such as radar load only) and a high value of total load to be served. Again, the Pareto solution, which satisfies this preference, is selected as the final solution.

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 [22]. 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 [23]-[24].

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

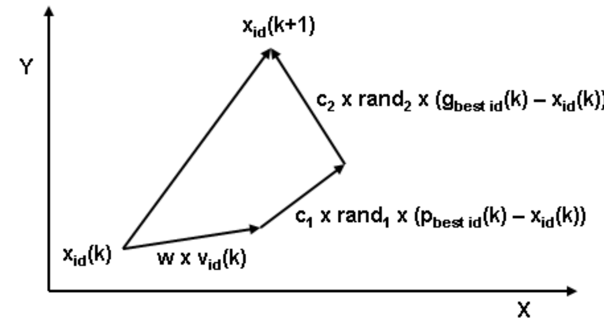

Fig. 2. Concept of changing a particle's position in two dimensions

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 [21]. 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.

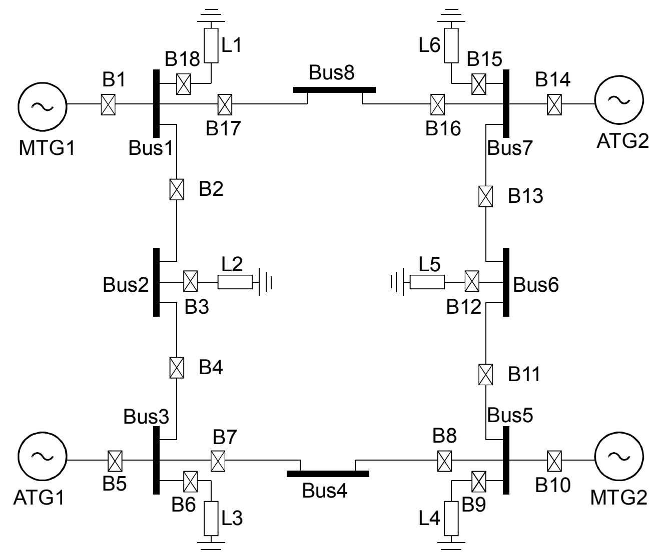

Fig. 3. Test system with six loads

A. *Matlab Based Case Study:*

For the test system presented in Fig. 3 (having six loads), the combinations of load magnitudes and load priorities considered in this paper are presented in Table I. For this 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. Table II corresponds to the fault scenario and outputs from the reconfiguration algorithm for case 1.

TABLE I
LOAD MAGNITUDE AND PRIORITIES FOR CASE 1

Load No.	L1	L2	L3	L4	L5	L6
Magnitude (MW)	2	20	5	2	20	4
Priority Weighting	10	1	4	10	1	5

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 51 MW. This definitely requires a load of at least 7 MW to be shed. For the sake of simplicity, generation reserve is not considered. Now, looking at the load magnitude and the priority weightings in Table I, it is clear that there are so many possible ways to shed the load of 7 MW or more. If the load L2 or L5 of 20 MW is shed, we can maximize the total priority weighing because both of them have the least priority. But, in that case total amount of load served by the system is much less than the generation capacity. Similarly, if load L3 and L4 are shed, the amount of load served by the system is maximum, but load L4 has a very high priority and we have to sacrifice that load which may not be desirable in some special situation. In order to take decision in this conflicting scenario, Pareto optimal solutions are obtained first by the SPPSO algorithm. Fig. 4 shows the Pareto front for this scenario. Since it is a discrete optimization problem, the Pareto front also consists of discrete points. Here three discrete operating points are obtained as the Pareto optimal solutions. Now, those solutions are passed through two questions which represent the user's preference regarding the operating point. In generalized form, the questions used in this paper are:

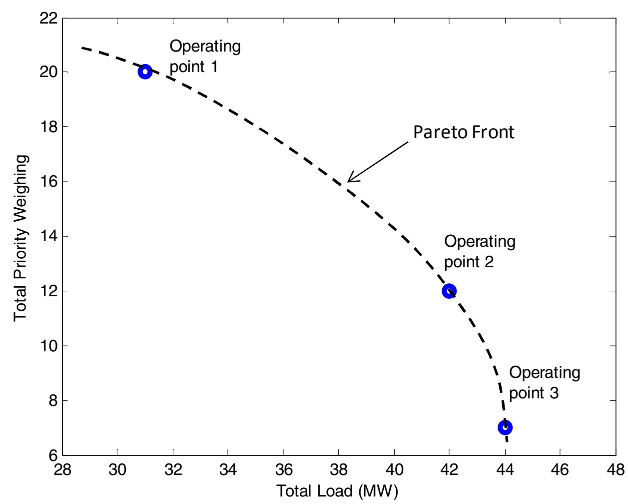

Fig. 4. Pareto optimal front for Case 1

Q_1. Is the critical amount of load powered?
Q_2. Is the critical load priority criteria met?

The critical minimum values of the load priority and the load magnitude are the part of the knowledge base. It can vary according to the mode of operation such as battle mode, normal mode, etc. The Pareto solution for which the answer to the above questions is affirmative, that solution is chosen as the final solution.

In this particular scenario, if the battle mode is represented by a critical value of priority equal to 20 and minimum amount of load to be powered equal to 30 MW, then the reconfiguration algorithm chooses the first operating point on the Pareto front. The load breaker configuration found by the algorithm suggests the shedding of L2 which is quite obvious from the Table I. Similarly, the other operating modes, the knowledge base corresponding to those modes and the suggestions for load shedding given by the reconfiguration algorithm are presented in Table II.

TABLE II
THE OUTPUT OF RECONFIGURATION ALGORITHM FOR CASE 1

Faulted Bus	Operating Mode	Critical Priority Requirement	Critical Load Requirement (MW)	Total Available Generation (MW)	Total Available Load (MW)	Possible Generator Breaker Matrix	Possible Load Breaker Matrix	Suggestion For Load-shedding
1	Mode 1	20	30	44	51	0111	011101	L5
	Mode 2	10	40	44	51	0111	010110	L3, L6
	Mode 3	6	44	44	51	0111	010011	L3, L4
2	NA	NA	NA	80	33	1000	101111	None

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 33 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.

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 III and this case is referred as case 2. Here the search space for the SPPSO algorithm becomes 2^8. With the application of fault at Bus 1 again, the available load becomes 54 MW and the available generation is 44 MW. This requires tripping of 10 MW of load. There are also few conflicting possibilities which require the extraction of Pareto optimal front. Fig. 5 shows the Pareto front obtained by the SPPSO algorithm. Once the Pareto front is obtained, the solutions are passed through the same questions mentioned before to choose the final solution. The knowledge base is varied to represent three different operating modes. It is observed that the reconfiguration algorithm finds out the optimal solution correctly for each operating mode. The results corresponding to each operating mode are summarized in Table IV.

TABLE III
LOAD MAGNITUDE AND PRIORITIES FOR CASE 2

Load No.	L1	L2	L3	L4	L5	L6	L7	L8
Magnitude (MW)	20	1	4	1	20	4	4	2
Priority Weighting	1	10	5	10	3	5	5	6

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. 6. 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. Here, to represent the operation mode, the critical priority weighing was set at 30 and the critical load to be powered was set at 30 MW (mode 1 of Table IV). 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 that load. In order to demonstrate the impact of reconfiguration on the loads, as well as on the entire system, a load L5 is selected. Before the occurrence of a fault, L5 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 L5 but it settled within two seconds. The dynamic variation of power and voltage at L5 load bus is shown in Figs. 7 and 8 respectively. Those are compared with the case where no reconfiguration is carried out. It is observed that the system drops to an unacceptable voltage (0.88 p.u.) within 16 seconds of the fault if no reconfiguration is carried out.

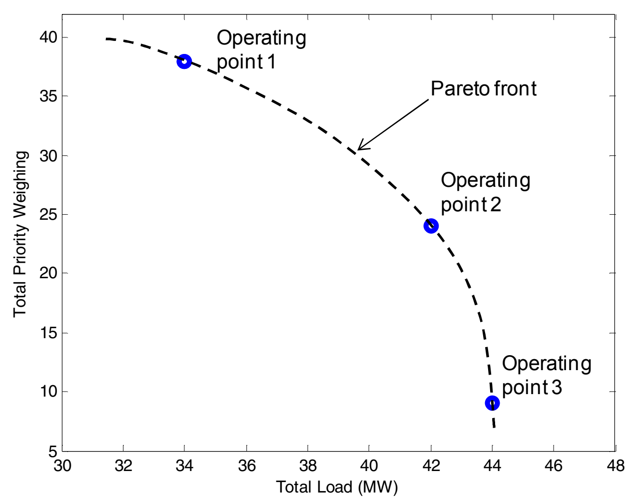

Fig. 5. Pareto optimal front for Case 2

TABLE IV
THE OUTPUT OF RECONFIGURATION ALGORITHM FOR CASE 2

Faulted Bus	Operating Mode	Critical Priority Requirement	Critical Load Requirement (MW)	Total Available Generation (MW)	Total Available Load (MW)	Possible Generator Breaker Matrix	Possible Load Breaker Matrix	Suggestion For Load-shedding
1	Mode 1	30	30	44	54	0111	10001010	L2,L3,L4,L6
	Mode 2	20	40	44	54	0111	11011000	L3,L6,L7
	Mode 3	6	44	44	54	0111	10101000	L2,L4,L6,L7

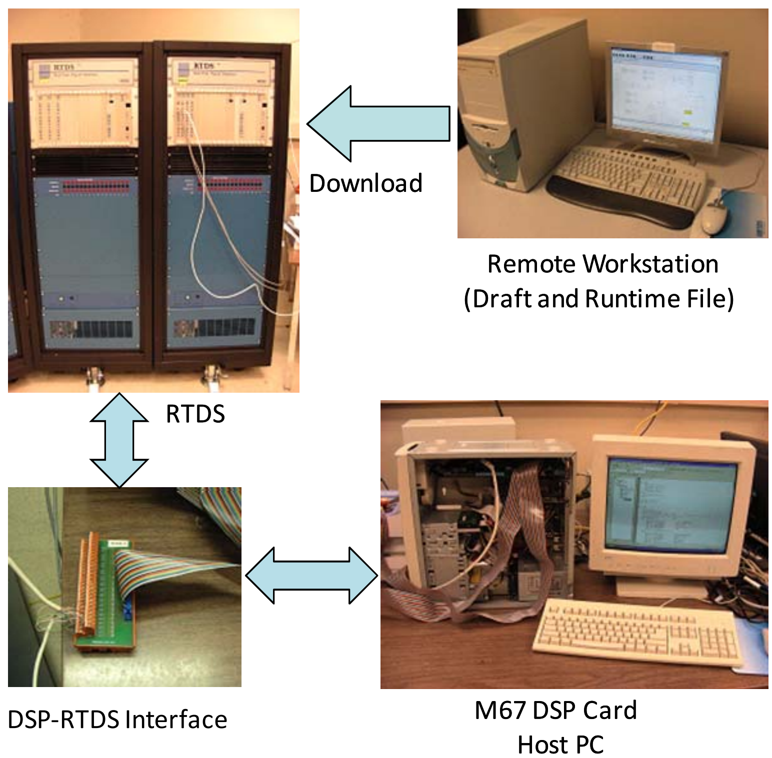

Fig. 6. Laboratory experimental setup

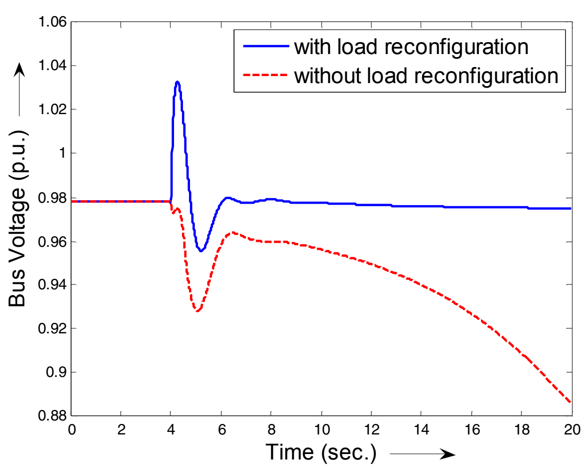

Fig. 7. Bus 6 voltage characteristics of load L5 post-fault

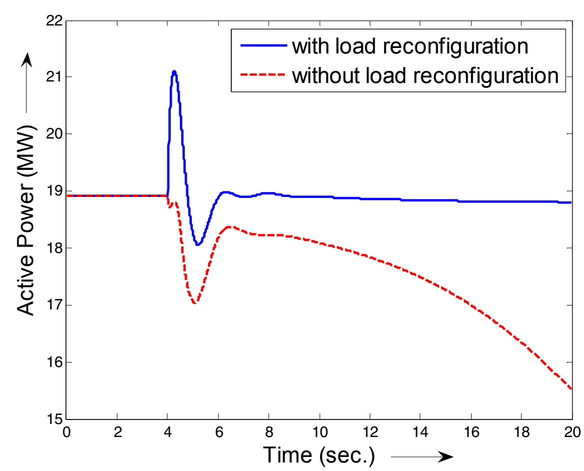

Fig. 8. Active power characteristics of load L5 post-fault

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. The Pareto optimal front is extracted from two conflicting objectives and the Pareto solutions are passed through two questions representing the user's preference regarding the mode of operation and thus the final solution is selected. Studies in Matlab and real-time environment are performed to illustrate the capability of the proposed reconfiguration strategy.

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. Intelligent fault identification is also another aspect of future research which can be integrated with the intelligent reconfiguration strategy.

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", *Intelligent System Applications in Power Systems*, 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] M. Tanaka, H. Watanabe, Y. Furukawa and T. Tanino, "GA-Based Decision Support System for Multicriteria Optimization", 1995, pp. 1556-1561.

[16] M. A. Abido, "A Niched Pareto Genetic Algorithm for Multiobjective Environment/Economic Dispatch", *Elsevier Journal of Electric Power and Energy Systems*, vol. 25, 2003, pp. 97-105.

[17] Carlos A. Coello Coello, M. S. Lechuga, "MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization", *Proceedings of the 2002 Congress on Evolutionary Computation*, vol. 2, May 2002, pp. 1051-1056.

[18] M. Reyes-Sierra, Carlos A. Coello Coello, "Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art", *International Journal of Computational Intelligence Research*, vol. 2, no. 3, 2006, pp. 287-308.

[19] U. Baumgartner, Ch. Magele and W. Renhart, "Pareto Optimality and Particle Swarm Optimization", *IEEE Transaction on Magnetics*, vol. 40, no. 2, March 2004, pp. 1172-1175.

[20] M. R. AlRashii and M. E. El-Hawary, "Pareto Fronts of the Emission-Economic Dispatch under Different Loading Conditions", *International Journal of Electric Power and Energy Systems Engineering* 1;2 Spring 2008, pp. 68-71.

[21] 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.

[22] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", *Proceedings of 1995 IEE Conference on Neural Networks*, vol. 4, pp. 1942-1948.

[23] 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 *IEEEExplore*.

[24] V. G. Gudise, G. K. Venayagamoorthy, "Comparison of Particle Swarm Optimization and Backpropagation as Training Algorithms for Neural Networks", *IEEE Swarm Intelligence Symposium*, Indianapolis, IN, USA, April 24-26, 2003, pp.110-117.