This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

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

Pinaki Mitra, *Student Member, IEEE*, Ganesh K. Venayagamoorthy, *Senior Member, IEEE*

***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 deficiencies for a long time, ultimately leading 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 the ship's power system. A fast, intelligent reconfiguration algorithm based on Small Population-based Particle Swarm Optimization (SPPSO) is presented in this paper. The reconfiguration of the electric ship's power system is formulated as a single objective as well as a multi-objective optimization problem. In the case of multi-objective optimization, the Pareto optimal solutions are obtained by SPPSO from two conflicting objective functions. From the Pareto set, the final solution is chosen depending on users' preferences regarding the mission of the navy ship. SPPSO is a variant of PSO having fewer numbers of particles and regenerating new solutions within the search space every few iterations. This concept of regeneration in SPPSO makes the algorithm fast and greatly enhances its capability. The strength of the proposed reconfiguration strategy is demonstrated on a Real-Time Digital Simulator (RTDS) environment.**

***Index Terms*— Dynamic reconfiguration, Electric ship, power system, Small Population-based Particle Swarm Optimization.**

I. INTRODUCTION

Reconfiguring distribution power networks is a well-known research area in power systems. Conventionally, it is viewed as a multi-objective optimization problem [1]. Heuristic search algorithms are the classical approach used to solve this distribution system reconfiguration problem [2]-[3]. Due to the stochastic nature of the problem, computational intelligence algorithms, such as genetic algorithms, particle swarm optimization, differential evolution, ant colony optimization, and a hybrid of an artificial immune system and ant colony optimization have been used by different researchers in [1] and [4]-[9], respectively. Generally, a common objective for the reconfiguration problem is to minimize the distribution system's losses. In some cases, as in [6], however, load balancing has been considered the primary objective. Along with loss minimization, some authors have considered additional objectives, such as minimizing voltage deviation and balancing transformer loads [1]. Numerous techniques have been adopted to solve the problem. In [2], a fast and simplified power flow program has been developed to determine the optimal flow pattern, which drives the heuristic algorithm. In [3], a heuristic algorithm has been proposed which starts with a meshed network with all switches in a closed state. Then, the switches are opened one by one, and the minimum loss configuration is calculated by running the load flow program for each configuration. A similar approach is considered in [4], where distributed generation has also been included into the problem's formulation. The only difference is that, instead of an exhaustive search, a genetic algorithm is used where each gene represents an open switch on the loops. In [5], along with a genetic algorithm, a fuzzy decision model has been added. The basic solution strategy remains the same. In [6], a load balancing index (LBI) has been minimized based on a binary particle swarm optimization algorithm. In [7], loss minimization is carried out on the subject to bus voltage and line current constraints with the help of the variable scaling hybrid differential evolution approach. Variable scaling helps to overcome the drawbacks of fixed and random scaling and solves the problem of mutation operator selection in hybrid differential evolution. In [8] and [9], the loss minimization objective is achieved by applying ant colony optimization in a hypercube framework.

However, a few basic differences exist between a normal distribution system and a naval shipboard's power system. The network loss in an electric ship is very small compared to that of a terrestrial distribution system; therefore, loss minimization is not an objective for reconfiguring an electric ship's power system. In navy ships, there are several emergency loads that must be served during battle conditions. Also, the ship's power system should reconfigure quickly so that the quality of power to those critical loads is maintained at the desired level at all times. Another consideration is that, in navy ships, there are concepts of 'missions;' the priorities of the loads vary according to the nature of the mission [10]. These

Manuscript received April 30, 2010. Part of this work was presented at the Industry Application Society Annual Meeting, 2009, Houston, USA. This work was supported in part by the US Office of Naval Research under the Young Investigator Program - N00014-07-1-0806 and the NSF CAREER grant ECCS # 0348221 both awarded to Dr. Venayagamoorthy.

Pinaki Mitra is with ABB, Chennai, India.

Ganesh K. Venayagamoorthy is with the Real Time Power and Intelligent Systems Laboratory, Missouri University of Science and Technology, Rolla, MO 65409 USA ((phone: 573-341-6641, fax: 573-341-4532, e-mail: pinakimitra@ieee.org, gkumar@ieee.org).

particularities of a ship's power system necessitate a simple, fast and intelligent reconfiguration strategy that can be implemented easily in real-time to produce the desired result. Dynamic reconfiguration of the ship's power system is an ongoing research area. In [11], a fast reconfiguration algorithm is proposed on a zone-based differential protection system. This algorithm has two consecutive search functions, a path search algorithm and a load shedding scheme based on load priorities for the path having a negative power balance. However, the authors do not present any real-time studies, making it difficult to predict how much time the algorithm will take to change the status of the breakers in a real system. This work is further developed in [12] and [13] by applying binary PSO and a genetic algorithm, respectively, for the load shedding scheme proposed in [11]. Generally, both PSO and GA work with a number of candidate solutions ('chromosomes' in GA and 'particles' in PSO). An increase in the number of potential solutions betters the exploration but eventually makes the algorithm slow and unfit for real-time applications. In addition to PSO and GA, agent-based reconfiguration strategies have been proposed in [14] and [15].

A new, simpler approach to reconfiguration which is fast enough to implement in real time without serious deterioration in voltage stability is presented in this paper. To enhance the algorithm's execution speed, the problem is first formulated as a single objective optimization problem, and the unique solution is obtained directly with Small Population-based Particle Swarm Optimization (SPPSO). However, this kind of formulation sometimes fails to address the few cases in which two conflicting objectives arise in practical situations. In those cases, it becomes necessary to formulate the problem as a multi-objective optimization problem. In a multi-objective framework, a set of Pareto optimal solutions are first extracted by 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 preferences corresponding to mission-specific requirements. Based on the response to those questions, the final solution is obtained. The concept of including user preferences in solving multi-objective optimization problems was introduced by Tanaka et. al. [16]. Their study utilized an offline process involving human interaction. However, in a ship's system, the speed of reconfiguration is the most important factor, so there is no scope for human participation. Hence, in this paper, the concept of user preference is included inside the algorithm itself in the form of a knowledge base.

The intelligent dynamic reconfiguration algorithm proposed in this paper for both the single objective and multi-objective formulations are first verified in Matlab and then implemented in real-time based on a Real Time Digital Simulator (RTDS) and a DSP.

The remainder of the paper is organized as follows: Section II discusses the detail of the proposed algorithm. The formulation of the problem as a multi-objective optimization problem and the concept of Pareto Optimality are presented in Section III. Section IV describes the principle of Small Population-Based Particle Swarm Optimization. Section V presents the test system and the results. Finally, the conclusion and future work are summarized in Section VI.

II. Intelligent Reconfiguration Algorithm

The ship's 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. In the typical structure shown in Fig. 1, 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 the remaining eight are path breakers. The status of the breakers can be either 'CLOSED' or 'OPEN,' hence; theoretically, 2^{20} breaker position possibilities exist. The breaker positions must also satisfy the condition

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

where P_{GEN} is the available generation at a particular time and P_{LOAD} , referred to as 'available load' in the remainder of the paper, is the amount of load to be powered at that point of time. 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 change simultaneously. Based on these changes, the reconfiguration strategy now searches for a new topology of the ship's power system so that it can supply the maximum number of critical loads with optimal generation. The objective functions for this problem can be formulated as follows:

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

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

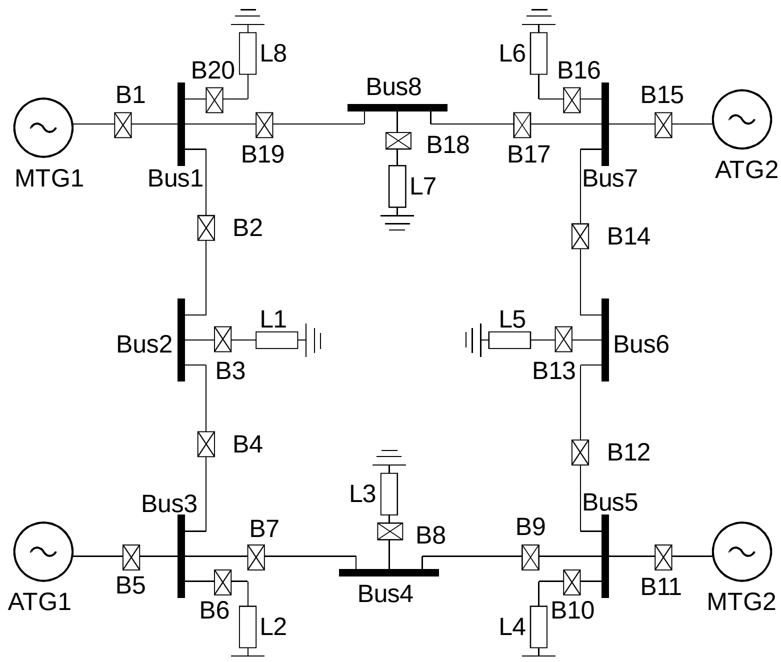

Fig. 1. Structure of ship's power system having eight buses (Bus1 to Bus8), four generators (MTG1, MTG2, ATG1 and ATG2), twenty breakers (B1 to B20), and eight loads (L1 to L8)

where p_i is the priority weighting associated with a load L_i for a particular mission and N is the total number of loads. A lower priority weighting signifies a lower priority. This type of formulation assumes that no conflicting objective exists; hence, the question of Pareto optimality does not arise. This is the simplest representation of the objective function and can successfully represent many common realistic scenarios. The proposed reconfiguration strategy consists of the following steps:

Step 1: First, the configuration of the ship's power system is represented by a vector of 20 binary elements, where '1' represents the 'CLOSED' and '0' represents the 'OPEN' status of the breakers, respectively. After obtaining the fault information, it updates the vector accordingly.

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

Step 3: From the updated generator and load breaker vectors, 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: 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 $M = 4$, as shown in Fig. 1. Hence, no intelligent algorithm is needed for this purpose. For $P_{GEN} < P_{LOAD}$, the proposed strategy carries out an optimal load shedding using the objective function in (2). The search space for this becomes 2^N, where $N = 8$, as in Fig. 1. However, real systems can have more loads, and with the addition of one load, the search space doubles. 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 an SPPSO algorithm [17] for maximizing (2).

III. Multi-objective Optimization and Pareto Optimality

When the objective function is represented as (2), the priority of the loads and the magnitudes of the loads are not considered separately. The implicit assumption behind (2) is that, while maximizing the product of the priority weighting and the magnitude of the loads, both are being maximized. In practice, however, this may not be so simple. For example, if, during load shedding, there is a very high priority load of 2 MW and a very low priority load of 20 MW, there are clearly two conflicting choices. If the load priority is important, then one will go for shedding the 20 MW load. However, this will not be the preferred choice if the situation needs the maximization of the total magnitude of the load to be served. Considering this practical conflicting situation, the objective represented by (2) can be split into two separate objectives as follows:

$$Max\left(\sum_{i=1}^{N} p_i\right) \tag{4}$$

and $$Max\left(\sum_{i=1}^{N} L_i\right) \quad (5)$$

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 multi-objective optimization problem, the objectives are to be achieved simultaneously, and they are formulated as [18]

$$\underset{x}{Minimize}\ f_i(x) \qquad i = 1,\ldots., N_{obj} \quad (6)$$

$$Subject\ to: \begin{cases} g_j(x) = 0 & j = 1,\ldots., M \\ h_k(x) \le 0 & k = 1,\ldots., K \end{cases} \quad (7)$$

Where f_i is the i^{th} objective function, x is a decision vector that represents a solution; and 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 multi-objective 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:

$$\forall i \in \{1,2,\ldots, N_{obj}\} : f_i(x_1) \le f_i(x_2) \quad (8)$$

$$\exists j \in \{1,2,\ldots, N_{obj}\} : f_j(x_1) < f_j(x_2) \quad (9)$$

If any element 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 solutions inside the search space is called the Pareto optimal set or the Pareto optimal front.

There are several methods to extract the Pareto optimal front in a multi-objective optimization problem [19]-[21], but in most of the techniques, the process of finding a 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 [22]:

$$f(x) = w_1 f_1(x) + w_2 f_2(x) \quad (10)$$

where,

$$w_1 + w_2 = 1 \quad (11)$$

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

In the case of a multi-objective framework, this Pareto optimal front extraction is carried out by an SPPSO algorithm. All the steps discussed in Section II remain the same except for a difference in Step 5. Previously, in the single objective formulation, the final solution was obtained by SPPSO in Step 5. In a multi-objective framework, instead of a unique solution, the entire set of Pareto optimal solutions are obtained by SPPSO. Therefore, in order to calculate the final solution, another step (Step 6) is added to the reconfiguration strategy.

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

IV. Small Population-Based PSO

A. *Conventional Particle Swarm Optimization Algorithm*

Particle swarm optimization is a population-based search algorithm that aims to replicate the motion of flocks of birds and schools of fish [23]. 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 [24]-[25].

Initially, a population of random solutions is considered. A random velocity is also assigned at which each individual particle begins flying within the search space. Also, each particle has a memory that keeps track of the previous best position of the particle and the corresponding fitness. This previous best value is called 'p_{best}.' Another value called 'g_{best}' 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 the i^{th} particle with d dimensions at instant k.

b) $x_{id}(k+1)$ is the position of the 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 particle's 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.

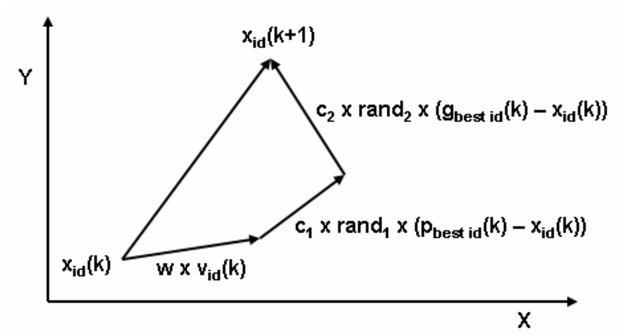

Fig. 2. Concept of changing a particle's position in two dimensions

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:

$$v_{id}(k+1) = w \cdot v_{id}(k) + c_1 \cdot rand_1 \cdot (p_{best\_id}(k) - x_{id}(k)) + c_2 \cdot rand_2 \cdot (g_{best\_id}(k) - x_{id}(k)) \quad (12)$$

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

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1) \quad (13)$$

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 because the higher the number of particles, the greater the exploration of the search space. However, as the number of particles increases, the memory requirement for the algorithm also increases, which often is not permissible in real-world applications of the algorithm with digital signal processors or microcontrollers. Also, the speed of convergence decreases significantly. In order to overcome these problems, an SPPSO algorithm was developed by Das and Venayagamoorthy in [22]. SPPSO starts with a small number of particles (generally around five), and, after a few iterations, replaces all the particles except for the global best with the same number of regenerated particles. In this method, because the PSO runs with a very small number of particles, the memory requirement decreases significantly. Also, because a new set of particles are introduced every few iterations, the chance of fixation to a local minima decreases and convergence is achieved much faster than in conventional PSO.

V. 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 results are broadly divided into two sections. The results with the single objective formulation are presented in Section A, and the results with the multi-objective formulation are presented in Section B. Within each section, the Matlab-based results are presented first, followed by the results achieved using real-time implementation.

A. Single Objective Formulation:

1) Matlab-Based Case Study:

For the test system presented in Fig. 1, a typical mission scenario (combination of load magnitudes and load priorities) is considered. This scenario is referred to as Case 1 (cruising) and is presented in Table I. For this case, a fault is created arbitrarily at a particular bus. The breaker status changes accordingly. The post-fault breaker status is then sent to the reconfiguration algorithm. The reconfiguration algorithm updates the breaker status vector. For the sake of convenience, it is assumed that all the breakers were in the 'CLOSED' state before the creation of the fault. Table II corresponds to the fault scenario and output from the reconfiguration algorithm for Case 1.

In the case of a fault at Bus 1 (fault scenario 1), breakers B1, B2, B19 and B20 (Fig. 1) are tripped. As a result, the generator MTG1 of 36 MW and load L8 of 2 MW are disconnected. Now, the total available generation is 44 MW, and the total available load is 66 MW. This requires shedding of at least 22 MW of load. For the sake of simplicity, generation reserve is not considered. Looking at the load priority weightings in Table I, it is clear that L3 and L5 have the least priority among the loads. However, if both of them are shed, the load priority is maximized, but not the total amount of load. As discussed earlier, in the single objective formulation, the product of priority weighting and the magnitude of the load should be maximized according to the cost function in (2) and not the priority alone. Therefore, along with L5, if either L2 or L6 is shed, the objective function in (2) is maximized. This was verified through an exhaustive search algorithm. The reconfiguration algorithm also correctly recommends these two possible solutions, as shown in Table II. In Table II, the generator and load breaker vectors are presented as binary strings. The elements from left to right of a generator breaker vector represent the status of G1, G2, G3 and G4 (Fig. 1), respectively. For example, a generator breaker vector 0111 means generator G1 is out of service, while G2, G3 and G4 remain connected. Similarly, for a load breaker vector, the elements from left to right represent the status of L1 to L8, consecutively. For example, a vector [10110110] means L2, L5 and L8 are disconnected and L1, L3, L4, L6 and L7 are connected.

To obtain better insight regarding the performance of the algorithm, a parameter sensitivity study of the algorithm based on a statistical analysis for fault scenario 1 is carried out. The performance of the SPPSO algorithm depends mostly on two parameters: a) number of particles in the swarm and b) number of iterations after which regeneration occurs. In Table III, the mean and standard deviation of the best fitness and the mean execution time of the algorithm over 50 trials are presented for a varying number of the two parameters mentioned above. The studies are carried out with an Intel ® Core ™ i7 1.73 GHz CPU and 3.05 GB of RAM. A standard deviation of zero is achieved only if the number of particles is greater than or equal to 4 and the number of iterations before regeneration is at least 7. If any of those two parameter values increases, the total number of fitness evaluations will also increase, and the time of execution of the algorithm will increase accordingly, which is not affordable in this type of application. Therefore, the minimum values of those two parameters (4 and 7, respectively) which produce a zero standard deviation are selected for the entire study. The average computation time over 50 trials for this case was 15.6 ms. The variation of the inverse of the cost function (in (2)) with a number of fitness evaluations for a random trial run is shown in Fig. 3. Increasing the size of the power system leads to an expected increase in the number of fitness evaluations required to attain a zero standard deviation, and the time of execution of the algorithm is expected to increase. However, there is no doubt that this algorithm will remain fast enough to find the global solution within a tolerable limit.

In fault scenario 2, a fault at Bus 2 is applied, but no generator is associated with the bus. For this fault, only load L1 of 20 MW is tripped. Since the total available generation is 80 MW and the total available load is only 48 MW, the reconfiguration algorithm recommends the tripping of both auxiliary generators (ATG1 and ATG2), each having a capacity of 4 MW, since the remaining generators are sufficient to serve the total available load.

TABLE I
LOAD MAGNITUDE AND PRIORITIES FOR CASE 1 (CRUISING)

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

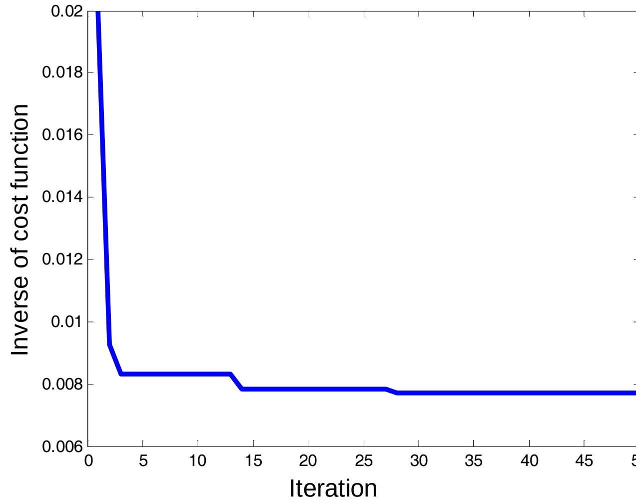

Fig. 3. Inverse of cost function vs. iteration curve for Case 1

2) RTDS-Based Case Study:

The model of an electric ship's 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 closely as that of a practical system. The real-time experimental setup is shown in Fig. 4. The breaker status signals from the RTDS are sent to the DSP. Using these signals, the reconfiguration algorithm implemented on the DSP recommends a new breaker status, if necessary.

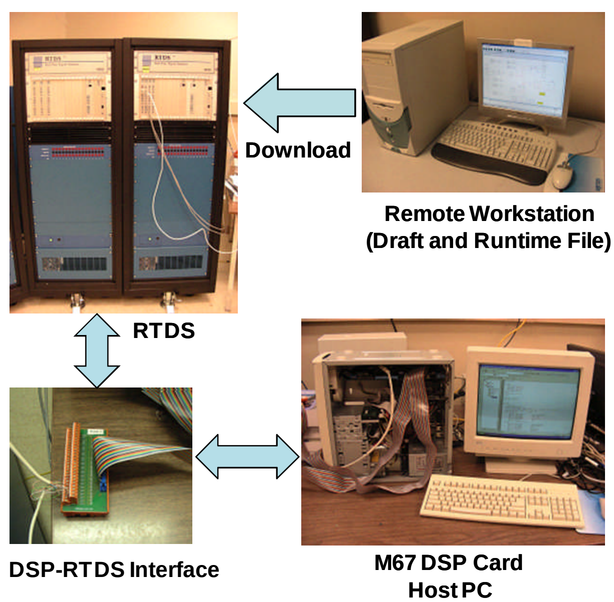

Fig. 4. Laboratory experimental setup

TABLE-II
THE OUTPUT OF RECONFIGURATION ALGORITHM FOR CASE 1 (CRUISING)

Faulted Bus	Total Available Generation (MW)	Total Available Load (MW)	Possible Generator Breaker Vector	Possible Load Breaker Vector	Suggestion For Load-shedding	Average Computation Time (ms)
1	44	66	[0111]	[10110110] or [11110010]	L2, L5 or L5, L6	15.6
2	80	48	[1010]	[01111111]	None	4.2

TABLE-III
STATISTICAL ANALYSIS OF SPPSO ALGORITHM FOR CASE 1 (CRUISING) (FAULT SCENARIO 1)

No. of particles	No. of iteration before every regeneration	Total no. of fitness evaluation	Mean best fitness over 50 trials	Standard deviation of best fitness over 50 trials	Mean time of execution of the algorithm (ms)
3	4	72	128.21	1.97e-4	7.7
	5	90	128.21	1.2e-4	7.8
	6	108	129.9	1.17e-4	9.8
	7	126	129.9	1.17e-4	11.7
4	4	96	128.2	9.56e-5	9.8
	5	120	129.9	6.25e-5	11.7
	6	144	129.9	2.19e-5	12.8
	7	168	130.0	0	15.6
5	4	120	129.9	1.63e-5	11.7
	5	150	129.9	1.18e-5	13.7
	6	180	129.9	1.17e-5	17.6
	7	210	130.0	0	19.5
6	4	144	129.9	1.24e-5	13.7
	5	180	129.9	4.46e-7	15.6
	6	216	129.9	2.2e-7	19.5
	7	252	129.9	0	21.5

The same load magnitude and priorities given in Table I are used. The same fault at Bus 1 is now applied from the RSCAD (an RTDS module) runtime window. The same results as observed in the Matlab study are obtained. Since the algorithm was calculated very quickly, the system had to remain overloaded for a short 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 the loads, as well as on the entire system, a high magnitude load L1 is selected. Before the occurrence of a fault, L1 was consuming 19 MW of power at a voltage of approximately 0.98 p.u. Post-fault, there was a transient in the voltage consumed by L1, but it finally settled down to a steady value of 1.0 p.u., which is very close to the pre-fault value. In this particular experiment, constant impedance-type loads have been considered, and no voltage control devices have been used. Therefore, with the fault, the subsequent isolation of the generator, and the reconfiguration of the loads, the entire system now moves to a new operating point. The voltages at each bus therefore change slightly. The dynamic voltage variation at the L1 load bus is shown in Fig. 5. Due to this change, the power consumed by the constant impedance loads also changes slightly. The dynamic variation of active power consumed by L1 is shown in Fig. 6. It is observed that the power settles down to a steady state value of 19.95 MW. The voltage and power characteristics are compared with the case in which no reconfiguration is carried out. It is observed that the system becomes unstable without load reconfiguration.

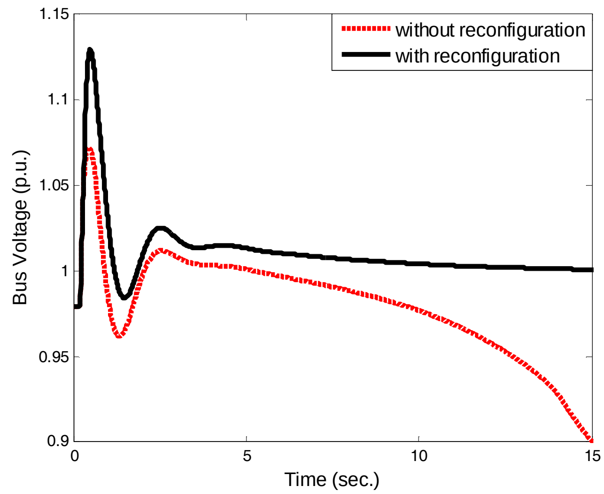

Fig. 5. Post-fault bus voltage characteristics of load L1 at Bus 2

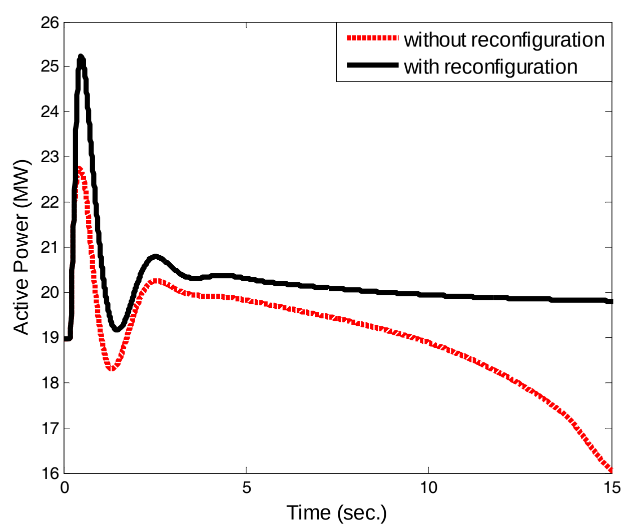

Fig. 6. Post-fault active power characteristics of load L1 at Bus 2

B. Multi-objective Formulation:

1) Matlab-Based Case Study:

A different set of magnitudes and priorities are considered for the multi-objective framework. Those values are listed in Table IV. A fault is created at Bus 1 as before. Thus, the generator MTG1 of 36 MW and load L8 of 2 MW are tripped. Now, the total available generation is 44 MW, and the total available load is 54 MW. This definitely requires a load of at least 10 MW to be shed. Here also, generation reserve is not considered. From the load magnitude and the priority weightings in Table IV, it can be calculated that there are 21 possible ways to shed a load of 10 MW or more. For example, if the load L1 of 20 MW is shed, we can maximize the total priority weighting because this load has the least priority. In that case, the total load amount served by the system is much less than the generation capacity. Similarly, if loads L2, L3, L4 and L6 are shed, the amount of load served by the system is maximized. However, the loads L2 and L4 both have very high priorities and must be sacrificed, which may not be desirable in some special situations. In order to make decisions concerning this conflicting scenario, Pareto optimal solutions are obtained first using the SPPSO algorithm. Fig. 7 shows the Pareto front for this scenario. The X-axis of Fig. 7 represents equation (5), and the Y-axis represents equation (4). 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 with varying values of w_1 and w_2 according to (11). Though w_1 and w_2 vary in increments of 0.1, only three non-dominant optimal solutions are obtained by SPPSO. Now, those solutions are passed through two questions representing user preferences regarding the mission. In a generalized form, the questions used in this paper are:

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 mean that at least that amount of load has to be served by the electric ship's power system in that particular mission mode. These two values corresponding to each mission are part of the knowledge base. Those values generally vary according to the type of mission, such as a battle mission, functional mission, cruising mission, etc. The Pareto solutions that satisfies those two conditions is chosen as the final solution.

In this particular scenario, the battle mission is represented by a critical value of priority equal to 35 and a minimum amount of load to be powered equal to 30 MW. Therefore, the reconfiguration algorithm chooses the first operating point on the Pareto front in that mission mode. The load breaker configuration found by the algorithm suggests the shedding of L1, which is obvious from Table IV. Similarly, the knowledge base corresponding to the other missions and the suggestions for load shedding given by the reconfiguration algorithm are presented in Table V. Here also, each case is run 50 times with the optimum SPSO parameters derived beforehand, and the average computation time is given in Table V.

2) RTDS-Based Case Study:

The same load magnitudes and priorities given in Table IV are used in the RTDS model. The fault at Bus 1 is now applied from the RSCAD runtime window as before. Here, to represent the functional mission, the critical priority weighting was set at 20, and the critical load to be powered was set at 40 MW (Table V). In order to demonstrate the impact of reconfiguration, a load L1 is selected. Before the occurrence of a fault, L1 was consuming 19.1 MW of power at a voltage of 0.983 p.u. Post-fault, there was a transient in the power consumed and the voltage of L1, but both settle down to respective steady values. The dynamic variation of voltage and power at the L1 load bus is shown in Figs. 8 and 9, respectively. Those are compared with the case in which no reconfiguration is carried out. It is again observed that the system destabilizes very slowly if the loads are not reconfigured.

TABLE-IV
LOAD MAGNITUDE AND PRIORITIES FOR CASE 2 (MULTI-OBJECTIVE SCENARIO)

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

TABLE-V
OUTPUT OF THE RECONFIGURATION ALGORITHM FOR CASE 2 (MULTI-OBJECTIVE SCENARIO)

Faulted Bus	Operating Mode	Critical Priority Requirement	Critical Load Requirement (MW)	Total Available Generation (MW)	Total Available Load (MW)	Possible Generator Breaker Vector	Possible Load Breaker Vector	Suggestion For Load-shedding	Average Computation Time (sec.)
1	***Battle***	35	30	44	54	[0111]	[01111110] (operating point 1)	L1	1.96
	Functional	20	40	44	54	[0111]	[11011000] (operating point 2)	L3,L6,L7	1.98
	Cruising	6	44	44	54	[0111]	[10101000] or [10001010] or [10001100] (operating point 3)	L2,L4,L6,L7 or L2,L3,L4,L6 or L2,L3,L4,L7	2.02

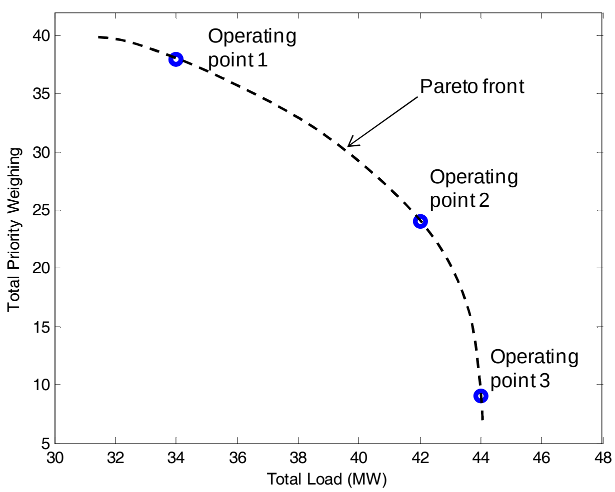

Fig. 7. Pareto optimal front for Case 2 (multi-objective scenario)

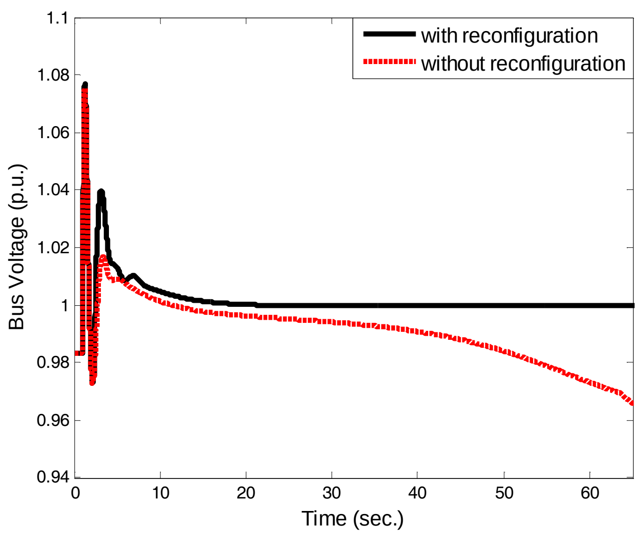

Fig. 8. Post-fault bus voltage characteristics of load L1 at Bus 2 (multi-objective scenario)

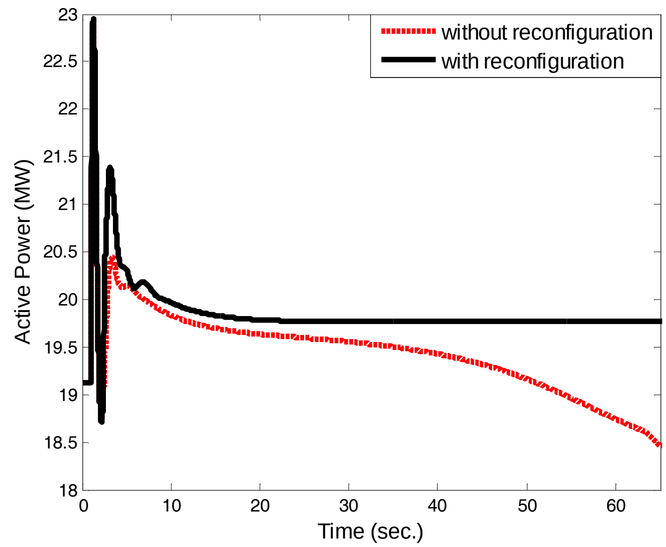

Fig. 9. Post-fault active power characteristics of load L1 at Bus 2 (multi-objective scenario)

VI. CONCLUSION

An intelligent dynamic reconfiguration strategy for an electric ship's power system has been presented in this paper. The problem is formulated both as a single objective and multi-objective optimization problem. Studies in Matlab and a real-time environment are performed to illustrate the capabilities of the proposed reconfiguration strategy.

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 this reconfiguration strategy is enhanced using a few simple, logical steps that greatly reduce the search space complexity. In the case of the multi-objective framework, the Pareto optimal front is extracted from two conflicting objectives, and the Pareto solutions are passed through two questions representing user preferences regarding the mission; thus, the final solution is selected.

In the future, more complex cases must be studied, such as the occurrence of multiple faults simultaneously on different buses, which may result in two or more islanded systems. This would require a path search algorithm to be included in the reconfiguration strategy. Intelligent fault identification is another aspect of future research that 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] Z. Ding, S. K. Srivastava, D. A. Cartes and S. Suryanarayanan, "Dynamic Simulation-Based Analysis of a New Load Shedding Scheme for a Notional Destroyer-Class Shipboard Power System", *IEEE Transaction on Industry Applications*, vol. 45, no. 3, May/June 2009, pp. 1166-1174.

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

[12] 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.

[13] 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.

[14] J. M. Solanki and N. N. Schulz, "Using Intelligent Multi-Agent Systems for Shipboard Power Systems Reconfiguration", *Proc. of the 13th International Conference on Intelligent Systems Applications to Power Systems*, ISAP 2005, pp. 212-214.

[15] 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.

[16] M. Tanaka, H. Watanabe, Y. Furukawa and T. Tanino, "GA-Based Decision Support System for Multicriteria Optimization", *IEEE International Conference on Systems, Man and Cybernetics*, 1995, pp. 1556-1561.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

[24] 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*, vol. 12, issue 2, 2008, pp. 171-195.

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