# A PSO Based Artificial Neural Network Approach for Short Term Unit Commitment Problem

AFTAB AHMAD*, AND AZZAM-UL-ASAR**

RECEIVED ON 14.01.2009 ACCEPTED ON 01.06.2009

## ABSTRACT

**UC (Unit Commitment) is a non-linear, large scale, complex, mixed-integer combinatorial constrained optimization problem. This paper proposes, a new hybrid approach for generating UC schedules using SI (Swarm Intelligence) learning rule based NN (Neural Network). The training data has been generated using DP (Dynamic Programming) for machines without valve point effects and using genetic algorithm for machines with valve point effects. A set of load patterns as inputs and the corresponding unit generation schedules as outputs are used to train the network. The NN fine tunes the best results to the desired targets. The proposed approach has been validated for three thermal machines with valve point effects and without valve point effects. The results are compared with the approaches available in the literature. The PSO (Particle Swarm Optimization) -ANN (Artificial Neural Network) trained model gives better results which show the promise of the proposed methodology.**

**Key Words:** **UC, ANN, , Dynamic Programming, PSO, SI.**

## 1. INTRODUCTION

The UC is an important and vital optimization task in the daily operational planning of modern power systems. UC is the problem of determining the optimal set of generating units within a power system to be used during the next 24-168 hours [1]. Traditionally the UC problem is to minimize the production costs (mainly fuel cost) and transition costs (start-up/shut-down costs) and is referred as the CBUC (Cost-Based Unit Commitment) problem [2-12]. In a CBUC problem, utilities have to produce power to satisfy their customers with the minimum production cost, fulfilling the condition that all power demand and reserve must be met. Since an improved UC schedule may save the electric utilities millions of dollars per year in production cost.

In general, the UC problem may be formulated as a non-linear, large scale, mixed-integer combinatorial optimization problem with both binary (unit status variable) and continuous (unit output power) variables [1-9]. The exact solution to the problem can be obtained only by the complete enumeration (Brute Force technique), often at the cost of prohibitive computational time for realistic power systems [1].

Researchers studied this complex problem for decades, and many conventional, meta heuristic and hybrid techniques have been developed [4]. The most popular conventional techniques to solve the UC problem have been the PL (Priority List), enumeration (brute force), DP,

* Professor, Department of Electrical Engineering, University of Engineering & Technology, Taxila.
** Professor, Department of Electrical Engineering, NWFP University of Engineering & Technology, Peshawar.

BB (Branch-and-Bound), IP/MIP (Integer/Mixed-Integer Programming) SF (Straight Forward), secant and LR (Lagrangian Relaxation) methods. All these methods have varying degrees of success with respect to solution time and optimality.

Recently, meta-heuristic techniques such as ANN, SA (Simulated Annealing), GA (Genetic Algorithm), EP (Evolutionary Programming), PSO, ACSA (Ant Colony Search Algorithm), TSA (Tabu Search Algorithm) and hybrid techniques combining different optimization techniques such as fuzzy- dynamic programming, LR-GA (Lagrange Relaxation Genetic Algorithm), LR-EP (Lagrange Relaxation Evolutionary Programming), PSO-SA, HPSO (Hhybrid Particle Swarm Optimization), DP=LR (Dynamic Programming Lagrange Relaxation), LR-PSO, TS-HPSO, MA (Memetic Algorithm), annealing AG, ACSA, DP-HNN (Hopfield Neural Network), MA-LR, SA-PSO, and EP-TSA have been used by many researchers to solve this difficult power-system-optimization problem due to their ability to solve UC problems more efficiently. These techniques accommodate more constraints and produce better solutions in an acceptable computation time. Despite the extensive work carried out, development of a new technique to solve the UC problem is still evolving. Research endeavors, therefore, have been focused on efficient, near-optimal UC algorithms, which can have reasonable storage and computational time requirements.

For the past several years, ANN methods received a great deal of attention. Ouyang, et. al. [13] utilizes neural networks to generate a pre-schedule according to the input load profile and then refines the schedule, where the commitment states of some of the units are uncertain, using a dynamic search. Sasaki, et. al. [14] explores the possibility of applying the Hopfield neural network to unit commitment. Sasaki, et. al. [15] utilizes the HNN in which a large number of inequality constraints are handled by the dedicated neural network instead of including them in the energy function. Once the states of generators are determined by the network, their outputs are adjusted according to the priority order in fuel cost per unit output. Yalcinoz, T., et. al. [16] presents a new mapping process and a computational method for obtaining the weights using a slack variable technique for handling inequality constraints. Wong, M.H., et. al. [17] used GA to evolve the weight and the interconnection of the neural network to solve the UC problem. Ouyang, Z., et. al. [18] proposes a multi-stage NN-expert system approach to achieve real time processing results. Huang, S.J., et. al. [19] proposes GA based neural network and DP to solve the UC problem. At the first stage a set of feasible generator commitment schedule is formulated by genetic-enhanced neural networks. In the second stage these pre-committed schedules are then optimized by the DP approach.

The advantage of using PSO algorithm over other techniques is that it can be computationally inexpensive, easily implemented, and does not require gradient information of the objective function but only its values. One of the first uses of PSO was for evolving neural network weights and, indirectly, to evolve the structure. Eberhart, Simpson, and Dobbins (1996) reported using PSO to replace the backpropagation learning algorithm. Particle swarm can directly evolve ANN by considering it as an optimization problem. Valle, Y. del, et. al. [20] presented a detailed overview of PSO including basic concepts, its variants and applications to power system. PSO applications as a training algorithm of ANN has been reported in [21-23]. Mohagheghi, et. al. [23] investigates the efficiency of PSO-based training of multilayer perceptron neural network.

In this paper, a new approach of PSO based ANN has been proposed to solve UC problem. The PSO algorithm is applied, to the neural network to obtain a set of weights that will minimize the error function. Weights are progressively updated until the convergence criterion is satisfied. We have decomposed the UC problem to discrete load level and formed small ANN models based on hourly load. The resultant small models get trained faster due to simple network structure and perform efficiently due to PSO based training strategy.

The paper is organized as follows: a brief overview of the UC problem and its formulation is given in Section 2, Section 3 and 4 addresses the discussion on ANN and PSO respectively. Implementation of proposed algorithm is given in Section 5. The experimental results of power system with 3 thermal units are presented and discussed in Section 6. The paper is finally concluded in the end.

## 2. UNIT COMMITMENT PROBLEM FORMULATION

Mathematical formulation of short term unit commitment problem is discussed as follows:

### 2.1 Objective Function

The principal objective is to prepare on/off schedule of the generating units in every sub-period (typically 1h) of the given planning period (typically 1 day or 1 week) in order to serve the load demand and reserve at minimum total production cost which includes fuel cost, start up cost, shut down cost, while meeting all unit, and system constraints. The following costs are considered.

#### 2.1.1 Fuel Cost

The quadratic approximation is the most widely used by the researchers, which is basically a convex shaped function. The operating fuel cost equation for unit i is described as:

$$F_i(P_{ih}) = a_i + b_i P_{ih} + c_i P_{ih}^2 \text{ (without valve point effects)} \tag{1}$$

The valve point effects produce a ripple, which is highly non-smooth and discontinuous. To take the effects of valve points a sinusoidal function is added to the convex cost function and represented as:

$$F_i(P_{ih}) = a_i + b_i P_{ih} + c_i P_{ih}^2 + |e_i \sin f_i (P_{ih\,min} - P_{ih})| \text{ (with valve point effects)} \tag{2}$$

#### 2.1.2 Start Up Cost

A simplified time-dependent start-up cost is mathematically given as a step function:

$$ST_{ih} = \text{h-cost}_i\text{: } T_i^{down}\ X_i^{off}(h) \le T_i^{down} + \text{c-s-hour}_i \quad \$/h \tag{3}$$

$$\text{c-cost}_i\text{: } X_i^{off}(h) > T_i^{down} + \text{c-s-hour}_i \quad \$/h \tag{4}$$

#### 2.1.3 Shut Down Cost

The typical value of the shut down cost is zero in the standard systems. This cost is considered as a fixed cost.

$$SD_{ih} = KP_{ih} \quad \$/h \tag{5}$$

where K is the incremental shut-down cost.

### 2.2 Constraints

The constraints that are considered and satisfied during the optimization process are as given:

#### 2.2.1 System Constraints or Coupling Constraints

Constraints that concern all the units of the system are called system or coupling constraints.

(i) System Power balance constraints or loading constraints:

$$\sum_{i=1}^{N} U_{ih} \left( P_{ih} \right) = D_h \qquad h = 1,2,\ldots\ldots H \tag{6}$$

(ii) System Spinning reserve requirements

$$\sum_{i=1}^{N} U_{ih} \left( P_{i\max} \right) \ge D_h + SR_h \qquad h = 1,2,\ldots\ldots H \tag{7}$$

#### 2.2.2 Unit Constraints or Local Constraints

Constraints that concern individual units are called unit constraints or local constraints.

(i) *Units Minimum and Maximum Generation Limits:* Each unit has generation range, which is represented as :

$$U_{ih}\ P_{imin} \le P_{ih} \le P_{imax}\ U_{ih} \qquad \text{for } i=1,2,\ldots N,\ h=1,2,\ldots H \tag{8}$$

(ii) Minimum up and Down Time Limits:

$$X_i^{on}(h) \ge T_i^{up},\ X_i^{off}(h) \ge T_i^{down} \quad \text{for } i=1,2,\ldots N,\ h=1,2,\ldots,H \tag{9}$$

(iii) *Units Status Restriction:* The unit status of each generating unit, Available/Not Available, Out aged/Must out, Must run, FOP (Fixed Output), Peakers/Gas Turbines must also be considered.

(iv) *Initial Status:* The initial status figure, if it is positive indicates the number of hours the unit is already up, and if it is negative indicates the number of hours the unit has been already down.

## 2. 3 UCP AS AN OPTIMIZATION PROBLEM

The objective function of the UCP (Unit Commitment Problem) is to minimize the total production cost. That is:

$$\text{Minimize TPC} = \sum_{h=1}^{H} \sum_{i=1}^{N} [\, F_i(P_{ih}) + ST_{ih} + SD_{ih} \,] \ \$/h \quad (10)$$

Subject to the system Equations (6-7) and unit constraints Equations (8-9).

## 3. ARTIFICIAL NEURAL NETWORK

An ANN in general is massively interconnected network of a large number of processing elements called neurons in an architecture, inspired by the brain. ANN exhibit characteristics such as mapping or pattern recognition, generalization, and fault tolerance. NN learn from examples and various learning laws exist of which supervised and unsupervised are popular. For a particular application the neural network is defined by its architecture and learning rule.

Back propagation is a systematic method of training multilayer feed forward artificial neural networks. It has been built on high mathematical foundation and has very good application potential. However, it has limitations. BP algorithm works well on small set of data and simple networks with few neurons. But when the problem under consideration has several parameters and many hidden layers, it becomes very difficult for backward propagation to minimize error. The ANN models using back propagation algorithm for training do not ensure convergence and hang in local minima and requires much longer training time. The neural network can be trained based on swarm intelligence learning rule [24].

ANN consists mainly of three layers: input layer, hidden layer and output layer. These layers are arranged in some way to have a ML (Multi Layer) feed-forward structure. A general network model consists of simple processing elements called neurons with adjustable parameters called weights. These neurons are arranged in a distinct layered topology and perform a biased weighted sum of their inputs and pass this activation level through a transfer relation (sigmoid function) to produce their output. Thus, the parameters of data flow from the input neurons, forwards through many hidden neurons, eventually reaching the output neurons. All of the layers are fully interconnected with each other by weights as shown in Fig. 1.

NNs learn by examples without using any programmed rules. The ability to learn through training is the most important feature of ANN. A supervised learning is given in Fig. 2. A NN uses a learning function to modify the variable connection weights at the inputs of each processing element according to some neural based algorithm. MLs of neurons with nonlinear activation functions allow the network to learn nonlinear and linear relationships between the input and output of the network. The training process in our network requires a set of examples of proper network behavior. The performance function of the NN is normally chosen to be the mean squared error for each pattern on the training set,

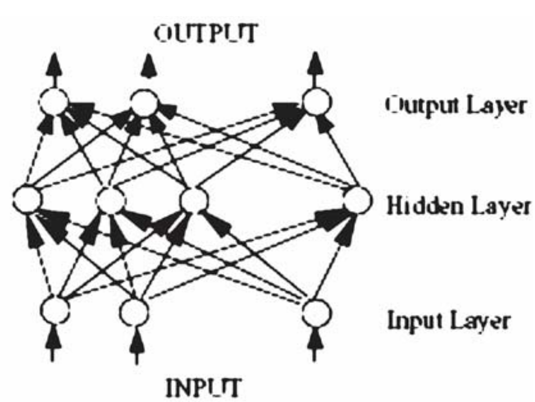


*FIG. 1. MULTILAYER FEED FORWARD NEURAL NETWORK ARCHITECTURE*

$$MSE = \frac{1}{P}\sum_{i=1}^{P}\left(t_{pi}-o_{pi}\right)^2 \qquad (11)$$

where $t_{pi}$ is the target at ith pattern, $o_{pi}$ is the network's output at ith pattern and P is number of NN pattern.

## 4. THE PARTICLE SWARM OPTIMIZATION

PSO presented by Kennedy and Eberhart [25-26] is one of the evolutionary computation technique based on the social behaviour of bird flocking and fish schooling. This is a population based stochastic global optimization technique. PSO has a population with random search solution. Each potential solution is represented as a particle in population called swarm. Since its introduction it has attracted lot of attention from the researchers around the world. PSO models problem as a set of n particles each representing a dimension of solution space. These particles move in solution space in search of optimal solution. The particles follow three principles as described by Kennedy [25] including evaluating: learning through self experience, Comparing: learning through comparative study and Imitating: learning through adapting the best trend. Each particle makes its own decision and influenced by its neighbors. As a particle try to imitate the best solution so far, in process may discover even better solution thus influence its neighbors. This process converges all particles to an optimal point similar to cultural trend in human society. The social sharing of information among the particles of a population may provide an evolutionary advantage which is the main idea behind the development of PSO algorithm. In PSO, only global best or local best gives information to others. It is one way information sharing mechanism. The evolution only looks for the best solution. PSO has been used to train multilayered neural network to solve variety of problems [27]. It has been used to optimize real and discrete functions which otherwise are difficult to solve.

The velocity and position update equations are given by:

$$v_{ij}(t+1) = w v_{ij}(t) + c_1 r_1\left[y_{ij}(t) - x_{ij}(t)\right] + c_2 r_2\left[\hat{y}_j(t) - x_{ij}(t)\right] \qquad (12)$$

$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)$ i=1, 2, 3,….s j=1,2 , 3,…..n (13)

where, $x_{i,j}$ is the current position of the particle at iteration j,w is inertia weight, $v_{i,j}$ is the current velocity of the particle at iteration j,$v_i(t+1)$ is the updated velocity of the particle, $y_i$ is the personal best position of the particle (every particle tries to adjust its velocity according to best positions ever visited that is stored in its memory), and $y^{\wedge}_i$ is an instance of $\hat{y}$ that is visited by the particle and yielded best positions currently found. $c_1$,$c_2$ are positive numbers and represent cognitive and social components respectively. $c_1$,$c_2$ controls the movement of the particle, $r_1$,$r_2$ are the uniform distribution numbers in the range [0,1], n be the dimension of the optimization problem, t the current instant and s is the swarm size.

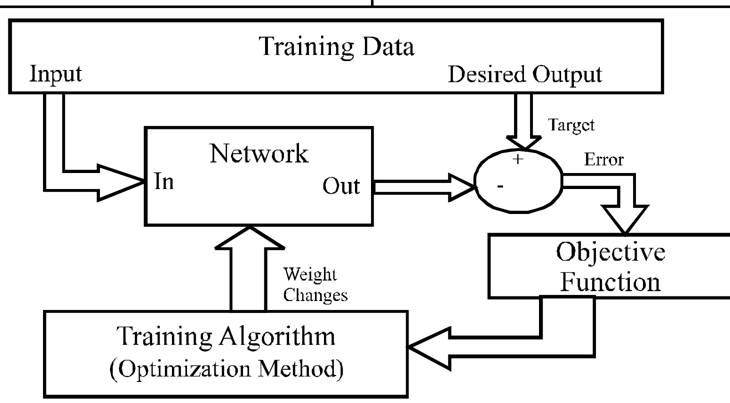


FIG. 2. SUPERVISED LEARNING

The original formula developed by Kennedy and Eberhart was improved by Shi and Eberhart [28-29] with the introduction of an inertia parameter w to prevent premature convergence and provides balance between local and global exploration. The performance of each particle is measured according to a predefined fitness function, which is related to the problem to be solved. The inertia weight w controls the impact of the previous histories of velocities on the current velocity, thus influencing the trade-off between global (wide-ranging) and local (nearby) exploration (exploitation) abilities of the "flying points". A larger inertia weight facilitates global exploration (searching new areas), while a smaller inertia weight tends to facilitate local exploration to fine-tune the current search area. Suitable choices of the inertia weight provide a balance between global and local exploration abilities and thus require less iterations to find the optimum solution. Although experimentation with the inertia weight is still in progress, it appears that a good general approach is to decrease the inertia weight linearly from 0.9-0.4 over 1,000 iterations. The following is the equation to obtain w:

$$W = W_{max} - \left(\frac{W_{max} - W_{min}}{iter_{max}}\right) \times iter \qquad (14)$$

where $w_{max}$ is the initial weight =0.9, $w_{min}$ is the final weight= 0.4, $iter_{max}$ is the maximum number of iterations, and iter is the current number of iteration

Clerc[30] introduced a constriction factor? that improves the ability of PSO and developed the following update rule.

The complete PSO formula is:

(15)

where

$$\chi = \frac{2}{\left|2 - \varphi - \sqrt{\phi^2 - 4\varphi}\right|} \qquad (16)$$

$$\varphi = c_1 + c_2,\ \varphi > 4 \qquad (17)$$

Bergh [31] showed that a dangerous condition arises when a particle position approaches closer to the global best position i.e. $x_i = y_i = \hat{y}$, the particle now will only depend on the term $wv_{ij}(t)$. In simple words when particle position coincides with the global best it will move only if w and v are non zero. Therefore, if the previous velocity of particle is close to zero all particles will stop moving. To solve this issue Bergh modified the equation of standard PSO by the GC (Guaranteed Convergence) PSO. The new equation used is represented by the global best particle as:

$$v_{\tau,j}(t+1) = -x_{\tau,j}(t) + \hat{y}_j(t) + wv_{\tau,j}(t) + \rho(t)(1-2r(t)) \qquad (18)$$

Where τ is the index of global best particle so that .
The term $-x_{\tau,j}(t+1)$ reset the particle position so that the particle only depends on $\hat{y}_j$. The term ρ(t)(1-2r(t)) generates random sample around $\hat{y}_j$ for searching better value of $\hat{y}_j$. The position update equation for the global best particle is given by:

$$x_{\tau,j}(t+1) = \hat{y}_j(t) + wv_{\tau,j}(t) + \rho(t)(1-2r(t)) \qquad (19)$$

$$\rho(t+1) = \begin{cases} 2\rho(t) & \text{if } N_S > S_c \\ 0.5\rho(t) & \text{if } N_{F,j} > f_c \\ \rho(t) & otherwise \end{cases}$$

The value of ρ(t) is changed after each iteration using the following rule:

(20)

Where $N_S$ number of successes denotes the number of improvements in the $\hat{y}$ and $N_F$ denotes the number of failures to improve the $\hat{y}$. A single failure is given by $\hat{y}(t) = \hat{y}(t-1)$. $S_c$ and $f_c$ are upper threshold values. The following two rules are required for implementation:

$$N_S(t+1) > N_F(t) \Rightarrow N_F(t+1) = 0$$

$$N_F(t+1) > N_F(t) \Rightarrow N_S(t+1) = 0$$

The optimal choice for the values of $S_c$ and $f_c$ depends upon the objective function. In high dimension search spaces it is difficult to obtain better values using random search in only a few iterations, so it is recommended to set $s_c$=15, $f_c$=5.

## 5. PROPOSED APPROACH

In this wok multilayer feed forward neural network with swarm intelligence learning rule has been programmed for tuning the power generation of machines. PSO trains the ANN by changing its weight such that MSE (Mean Square Error) for the training set of an ANN model is reduced. It is a feedback process which runs until the batch error falls under an acceptable range or iteration cutoff threshold is reached. In the process of training ANN, particle position corresponds to the weights in the network. The fitness function corresponds to the MSE of the network. So each particle represents a possible solution network. Particles cooperate with each other to find an optimal network with minimal MSE.

### 5.1 Input and Output of the PSO-ANN Model

The input to the network is the forecasted load demand as shown in Fig. 3. For a three machine system, in the ANN model three untrained inputs ($P_1$, $P_2$, $P_3$, the power generation of the machines) and three trained outputs $P_1$, $P_2$, $P_3$ have been taken. For proper training of ANN a pair of load as input and their corresponding generation schedules as output are prepared off-line by dynamic programming as shown in Fig. 4 and are stored in a data base. Each pair is referred as input/output database pair. The networks is trained with input/ output data pattern with 3 neurons as input, 2 neurons for hidden layers and 3 neurons for output layer. Total we have 24 networks for 24 hour duration each representing a hourly schedule h. Each network has been trained separately over hourly load data.

The target has been taken as the best solution attained so far. The percentage error is calculated as:

((target - network output) / target)) x 100

### 5.2 Scaling of the Input and Output Data

The network input and output data will have different ranges if actual hourly load data is used so it was normalized to fall within the range [0, 1], to avoid convergence problems during the training process.

### 5.3 Training Process

The training process of an ANN model by PSO approach has the following algorithm:

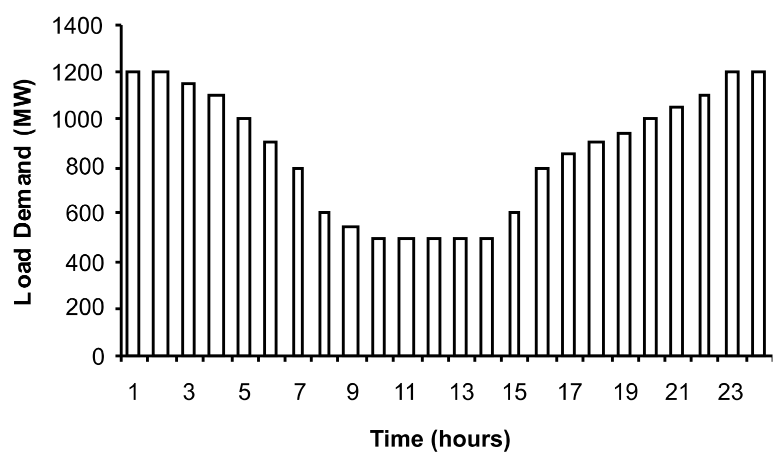


*FIG. 3. LOAD PATTERNS FOR TRAINING*

```
Initialize swarm with random velocity and position
Repeat
        for each particle i in swarm do
                if MSE( current network ) < MSE (personal
                best network) then
                        Personal best network = current
                        network
                end if
                if MSE (current network)< the MSE (global
                best network) then
                        Global best network = current
                        network
                end if
        end for
        Update velocity and position of each particle according
        to equations (12) and (13)
Until stop criteria (epoch < 30000  or MSE (global best network)
< 1.0e-12) being satisfied
```

The perception training algorithm is a form of supervised learning algorithm where the weights and biases are updated to reduce errors whenever the network output does not match the desired values. The load patterns and its corresponding feasible schedules are obtained using dynamic programming technique in a training set for the ANN. Most of the time is spend on off-line training of the network. With trained network the on-line operation time is very short.

The terminating criteria is the maximum number of epochs and the maximum mean square error.

The PSO model parameters used in the analysis are as:

Number of particles =20, scaling constants, $c_1=c_2=1.4962$, Constriction factor $\chi$=0.7298, the range for individual particle position $2\leq x\leq -21$, the range for individual dimension of particle velocity, is v_max=k*x_max, x_max=21, k=0.1 and is equal to $2.1\leq x\leq -2.1$, Maximum Epoch=30000, Maximum Error=1.0e-12, max_memory=30, success_counter $s_c$ =15, fail_counter $f_c$ =5.

## 6. EXPERMENTAL RESULTS

Three machine systems was investigated for the validation of the proposed approach. The data of the units are given in Table 1. It relate these parameters/variables with the table and shows the unit number (1-3), parameters of the fuel cost curve (a, b and c), constants from the valve point effect (e and f), minimum generation capacity ($P_{min}$), and maximum generation capacity ($P_{max}$) of each unit.

The results have been tabulated in Tables 2-4. Table 2 gives the complete unit commitment schedule, the target outputs and the results obtained by proposed SI-ANN and BP-ANN for units without valve point effects. The training set covers 24 hours load data. We used 24 networks for 3 units system each having architecture of 3:2:3. The training set contained 20 patterns for each hour. Total number of training patterns is 480.

The results of SI learning ANN has been compared with back propagation learning rule ANN. The Figs. 4-6 show the plot of mean square error against number of epochs for both learning rules for three test cases. In all cases the back propagation ANN jumps to lower value, and then iterates in very narrow band and does not hit the target values in 30,000 epochs. Where as for SI-ANN the decay in mean square error is gradual as compared to back propagation ANN, but it hits the targets in the given number of 30,000 epochs. The Figs. 4-6 show the plot for first 100 epochs, as for higher epochs the error remains low.

The comparison of variation of percentage error for operating fuel cost obtained from both BP-ANN and SI-ANN is shown in Fig. 7. But in case of SI-ANN this error is almost zero indicating that the fuel cost approaches the target and hence indicates the strength of this approach.

**TABLE 1. GENERATOR DATA FOR 3 UNIT SYSTEM**

| UNIT No. | $P_{min}$ (MW) | $P_{max}$ (MW) | a | b | c | e | f |
|---|---|---|---|---|---|---|---|
| 1 | 150 | 600 | 561 | 7.92 | 0.001562 | 300 | 0.0315 |
| 2 | 100 | 400 | 310 | 7.85 | 0.00194 | 200 | 0.0420 |
| 3 | 50 | 200 | 93.6 | 9.564 | 0.00578 | 150 | 0.0630 |

In the unit commitment problem the total number of combinations we need to try each hour is $2^N$-1, where N is the number of units. For a total period of H interval, the maximum number of possible combinations is $(2^N-1)^H$, and hence for three machine system it is $1.915 \times 10^{20}$.

Twenty samples in each case consisting of one to three input(s) for 24 hours load is provided to the neural network. The neural network produces twenty outputs consisting of power generations, P1-P3. The total production cost for all samples and percentage error with respect to target production cost have been calculated.

We compare the results of total operating fuel cost with the other two approaches available in the literature. The results of our approach are 0.234% less as compared with Hopfield neural network and 0.192 percentage less than the GA as mentioned in Table 3. Millions of dollars per year can be saved by a small percentage reduction in fuel cost. The results obtained by our approach are better than Hopfield neural network and genetic algorithm which shows the promise of the proposed approach.

The accurate and more practical solution is obtained by inclusion of the valve point effects in the fuel cost function model. The generating units with multi-valves have nonconvex and non-linear input-output characteristics. The fuel cost is high, as the load on the unit increases the opening of the valves rapidly increases the throttling losses.

**TABLE 2. UNIT COMMITMENT SCHEDULE AND THE OUTPUTS OBTAINED BY PROPOSED SI-ANN AND BP-ANN FOR MACHINES WITHOUT VALVE POINT EFFECTS**

| Hour | UC Sch. | Target Output | | | Load (MW) | Output Obtained by the Proposed SI ANN | | | Output Obtained by BP ANN | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Unit-1 MW | Unit-2 MW | Unit-3 MW | | Unit-1 MW | Unit-2 MW | Unit-3 MW | Unit-1 MW | Unit-2 MW | Unit -3 MW |
| 1 | 111 | 600.00 | 400.000 | 200.000 | 1200 | 600.000 | 400.000 | 200.000 | 599.999 | 399.999 | 199.999 |
| 2 | 111 | 600.00 | 400.000 | 200.000 | 1200 | 600.000 | 400.000 | 200.000 | 599.999 | 399.999 | 199.999 |
| 3 | 111 | 600.00 | 400.000 | 150.000 | 1150 | 600.000 | 400.000 | 150.000 | 599.999 | 399.999 | 149.999 |
| 4 | 111 | 600.00 | 400.000 | 100.000 | 1100 | 600.000 | 400.000 | 100.000 | 599.999 | 399.999 | 99.999 |
| 5 | 110 | 600.00 | 400.000 | 0 | 1000 | 600.000 | 400.000 | 0 | 599.984 | 400.000 | 0 |
| 6 | 110 | 500.00 | 400.000 | 0 | 900 | 500.000 | 400.000 | 0 | 499.998 | 399.950 | 0 |
| 7 | 110 | 433.18 | 366.820 | 0 | 800 | 433.180 | 366.820 | 0 | 433.192 | 366.849 | 0 |
| 8 | 100 | 600.00 | 0 | 0 | 600 | 600.000 | 0 | 0 | 599.999 | 0 | 0 |
| 9 | 100 | 550.00 | 0 | 0 | 550 | 550.000 | 0 | 0 | 549.999 | 0 | 0 |
| 10 | 100 | 500.00 | 0 | 0 | 500 | 500.000 | 0 | 0 | 499.999 | 0 | 0 |
| 11 | 100 | 500.00 | 0 | 0 | 500 | 500.000 | 0 | 0 | 499.999 | 0 | 0 |
| 12 | 100 | 500.00 | 0 | 0 | 500 | 500.000 | 0 | 0 | 499.999 | 0 | 0 |
| 13 | 100 | 500.00 | 0 | 0 | 500 | 500.000 | 0 | 0 | 499.999 | 0 | 0 |
| 14 | 100 | 500.00 | 0 | 0 | 500 | 500.000 | 0 | 0 | 499.999 | 0 | 0 |
| 15 | 100 | 600.00 | 0 | 0 | 600 | 600.000 | 0 | 0 | 599.999 | 0 | 0 |
| 16 | 110 | 433.18 | 366.820 | 0 | 800 | 433.180 | 366.820 | 0 | 433.192 | 366.849 | 0 |
| 17 | 110 | 460.88 | 389.120 | 0 | 850 | 460.880 | 389.120 | 0 | 460.880 | 389.118 | 0 |
| 18 | 110 | 500.00 | 400.000 | 0 | 900 | 500.000 | 400.000 | 0 | 499.998 | 399.950 | 0 |
| 19 | 110 | 550.00 | 400.000 | 0 | 950 | 550.000 | 400.000 | 0 | 550.009 | 399.969 | 0 |
| 20 | 110 | 600.00 | 400.000 | 0 | 1000 | 600.000 | 400.000 | 0 | 599.984 | 400.000 | 0 |
| 21 | 111 | 600.00 | 400.000 | 50.000 | 1050 | 600.000 | 400.000 | 50.000 | 599.999 | 399.999 | |
| 22 | 111 | 600.00 | 400.000 | 100.000 | 1100 | 600.000 | 400.000 | 100.000 | 599.999 | 399.999 | 99.999 |
| 23 | 111 | 600.00 | 400.000 | 200.000 | 1200 | 600.000 | 400.000 | 200.000 | 599.999 | 399.999 | 199.999 |
| 24 | 111 | 600.000 | 400.000 | 200.000 | 1200 | 600.000 | 400.000 | 200.000 | 599.999 | 399.999 | 199.999 |

**TABLE 3. TOTAL BEST COST COMPARISON WITH OTHER APPROACHES FOR MACHINES WITHOUT VALVE POINT EFFECTS**

| Technique | Total Best Cost | Reduction in Operating Fuel Cost ($/day) | Percentage Reduction | Saving Per Year (M$/year) |
|---|---|---|---|---|
| Hopfield Neural Network [32] | 202106.68 | 474.60 | 0.234826 | 0.173 |
| Genetic Algorithm [33] | 202021.36 | 389.28 | 0.192692 | 0.142 |
| Proposed Approach | 201632.08 | - | - | - |

**TABLE 4. UNIT COMMITMENT SCHEDULE AND ECONOMIC DISPATCH FOR MACHINES WITH VALVE POINT EFFECTS (PSO-ANN APPROACH)**

| Hour | UC sch. | Power Output of Unit-1 (MW) | Power Output of Unit-2 (MW) | Power Output of Unit-3 (MW) | Load (MW) | Operating Fuel Cost of Unit-1 ($/h) | Operating Fuel Cost of Unit-2 ($/h) | Operating Fuel Cost of Unit-3 ($/h) | Total Production Cost ($) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 111 | 600.00 | 400.00 | 200.00 | 1200 | 6175.105 | 3767.125 | 2241.383 | 12183.613 |
| 2 | 111 | 600.00 | 400.00 | 200.00 | 1200 | 6175.105 | 3767.125 | 2241.383 | 12183.613 |
| 3 | 111 | 600.00 | 400.00 | 150.00 | 1150 | 6175.105 | 3767.125 | 1660.772 | 11603.002 |
| 4 | 111 | 600.00 | 400.00 | 100.00 | 1100 | 6175.105 | 3767.125 | 1109.061 | 11051.291 |
| 5 | 110 | 600.00 | 400.00 | 0 | 1000 | 6175.105 | 3767.125 | 0 | 9942.230 |
| 6 | 110 | 500.00 | 400.00 | 0 | 900 | 5211.370 | 3767.125 | 0 | 8978.495 |
| 7 | 110 | 433.20 | 366.80 | 0 | 800 | 4429.946 | 3645.997 | 0 | 8075.943 |
| 8 | 100 | 600.00 | 0 | 0 | 600 | 6175.105 | 0 | 0 | 6175.105 |
| 9 | 100 | 550.00 | 0 | 0 | 550 | 5399.592 | 0 | 0 | 5399.592 |
| 10 | 100 | 500.00 | 0 | 0 | 500 | 5211.370 | 0 | 0 | 5211.370 |
| 11 | 100 | 500.00 | 0 | 0 | 500 | 5211.370 | 0 | 0 | 5211.370 |
| 12 | 100 | 500.00 | 0 | 0 | 500 | 5211.370 | 0 | 0 | 5211.370 |
| 13 | 100 | 500.00 | 0 | 0 | 500 | 5211.370 | 0 | 0 | 5211.370 |
| 14 | 100 | 500.00 | 0 | 0 | 500 | 5211.370 | 0 | 0 | 5211.370 |
| 15 | 100 | 600.00 | 0 | 0 | 600 | 6175.105 | 0 | 0 | 6175.105 |
| 16 | 110 | 433.20 | 366.80 | 0 | 800 | 4429.946 | 3645.997 | 0 | 8075.943 |
| 17 | 110 | 460.80 | 389.20 | 0 | 850 | 4649.409 | 3740.631 | 0 | 8390.040 |
| 18 | 110 | 500.00 | 400.00 | 0 | 900 | 5211.370 | 3767.125 | 0 | 8978.495 |
| 19 | 110 | 550.00 | 400.00 | 0 | 950 | 5399.592 | 3767.125 | 0 | 9166.717 |
| 20 | 110 | 600.00 | 400.00 | 0 | 1000 | 6175.105 | 3767.125 | 0 | 9942.230 |
| 21 | 111 | 600.00 | 400.00 | 50.00 | 1050 | 6175.105 | 3767.125 | 586.250 | 10528.480 |
| 22 | 111 | 600.00 | 400.00 | 100.00 | 1100 | 6175.105 | 3767.125 | 1109.061 | 11051.291 |
| 23 | 111 | 600.00 | 400.00 | 200.00 | 1200 | 6175.105 | 3767.125 | 2241.383 | 12183.613 |
| 24 | 111 | 600.00 | 400.00 | 200.00 | 1200 | 6175.105 | 3767.125 | 2241.383 | 12183.613 |
| The Total Operating Fuel Cost for 24 Hours is $ 208325.261 | | | | | | | | | |

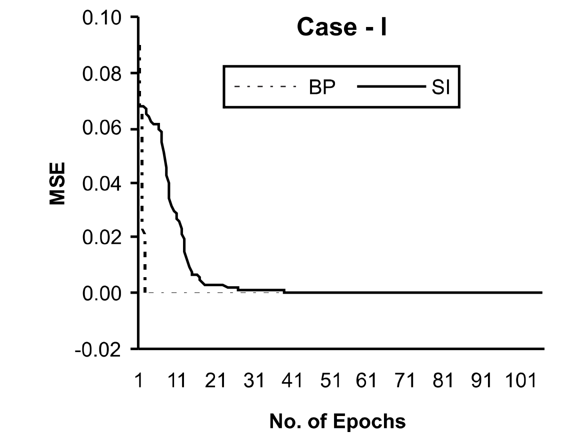


*FIG. 4. MEAN SQUARE GRAPH FOR A LOAD OF 500MW*

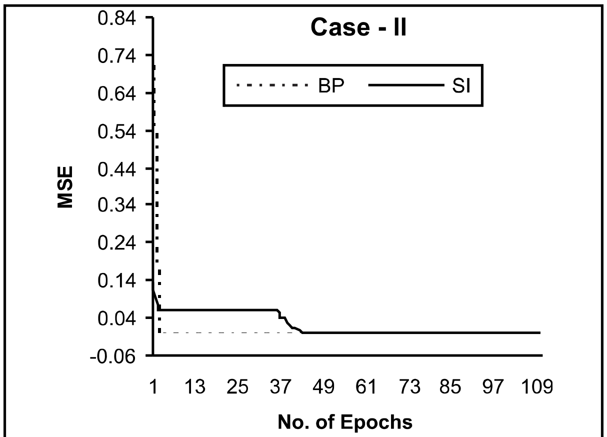


*FIG. 5. MEAN SQUARE ERROR GRAPH FOR A LOAD OF 800MW*

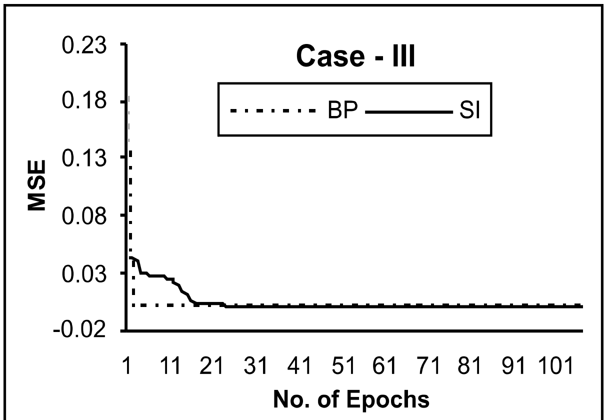


*FIG. 6. MEAN SQUARE ERROR GRAPH FOR A LOAD OF 850MW*

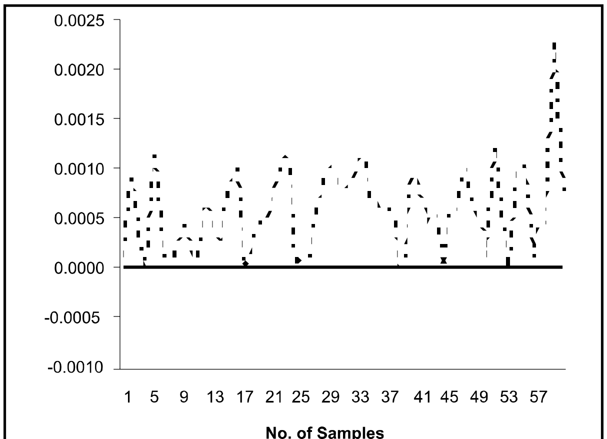


*FIG. 7. ABSOLUTE PERCENTAGE ERROR GRAPH FOR OPERATING FUEL COST*

## 7. CONCLUSION

This paper has presented a new approach for the solution of the unit commitment problem. Brute force (enumeration) is the best solution methodology for the unit commitment problem. This is the exact method but takes a long time to reach the solution. Proposed PSO-ANN approach gives the solution to the problem quickly and efficiently. This approach is simple as compared to the other approaches in terms of complexity. Swarm intelligence learning based model gives better results. For the systems consisting of large number of machines, the modular approach will be experimented.

## 8. NOTATION

The following notations are used in this paper:

| | |
|---|---|
| $P_{ih}$ | real power output of unit i at hour h,(MW) |
| $U_{ih}$ | the on/off status of unit i at hour h. $U_{ih}$=0 when OFF, $U_{ih}$=1 when ON |
| $F_i(P_{ih})$ | fuel cost function or fuel cost rate of unit i (\$/h) |
| $a_i,b_i,c_i$ | positive fuel cost coefficients of unit i measured in \$/h, \$/MW h and \$/MW$^2$ h, respectively, |
| $e_i,f_i$ | constants from the valve point effect of the unit i. |
| $X_i^{on}(h)$ | duration during which the unit i is continuously ON,(h) |

$X_i^{off}(h)$ duration during which the unit i is continuously OFF,(h)

$h\text{-}cost_i$ hot start cost of unit i ($),

c-costi cold start cost of unit i ($),

$c\text{-}s\text{-}hour_i$ cold start time of unit i (h),

N the number of units

H the number of hours, (24 h)

$D_h$ load demand at hour h, (MW)

$SR_h$ spinning reserve at hour h, (MW)

$T_i^{up}$ minimum up time of unit i, (h)

$T_i^{down}$ minimum down time of unit i, (h)

TPC total production cost, ($/h)

$ST_{ih}$ : start up cost of unit i in hour h, ($/h)

$SD_{ih}$ shut down cost of unit i in hour h, ($/h)

$P_{imin}$ minimum generation limit of unit i, (MW)

$P_{imax}$ maximum generation limit of unit i, (MW)

## ACKNOWLEDGEMENTS

The authors are indebted to the University of Engineering and Technology, Taxila, Pakistan, and Higher Education Commission, Islamabad, Pakistan, for providing support in completing the present research work through Endowment Fund Project. The technical support of Dr. Aftab Ahmad (Late), Director Private Power, NPCC, WAPDA, Islamabad in undertaking this work is specially acknowledged.

## REFERENCES

[1] Wood, A.J., and Wollenberg, B.F., "Power Generation, Operation and Control", John Wiley & Sons, Inc, NY, 1984.

[2] Xiaomin, B., and Shaidehpour, S.M., "Extended Neighbourhood Search Algorithm for Constrained Unit Commitment", Electric Power Energy System, Volume 19, No,. 5, pp. 349-356, 1997.

[3] Sheble, G.B., and Maifeld, T.T., "Unit Commitment by Genetic Algorithm and Expert System", Electric Power System Research, Volume 30, pp. 115-121, 1994.

[4] Yamin, H.Y., "Review on Methods of Generation Scheduling in Electric Power Systems", Electric Power System Research, Volume 69, pp. 227-248, 2004.

[5] Bakirtzis, A.G., and Zoumas, C.E., "Lambda of Lagrangian Relaxation Solution to Unit Commitment Problem", IEE Proceedings on Generation Transmission and Distribution, Volume 147, pp. 131-136, 2000.

[6] Ma, X., El-Keib, A.A., Smith, R.E., and Ma, H., "A Genetic Algorithm Based Approach to Thermal Unit Commitment of Electric Power Systems", Electric Power System Research, Volume 34, pp. 29-36, 1995.

[7] Orero, S.O., and Irving, M.R., "A Genetic Algorithm for Generator Scheduling in Power Systems", Electric Power Energy Systems, Volume 18, No. 1, pp. 19-26, 1996.

[8] Kazarlis, S.A., Bakirtzis, A.G., and Petridis, V., "A Genetic Algorithm Solution to the Unit Commitment Problem", IEEE Transactions on Power Systems, Volume 11, No. 1, pp. 83-92, February, 1996.

[9] Mantawy, A.H., Abdel-Magid, Y.L., and Selim, S.Z., "Unit Commitment by Tabu Search", IEE Proceedings on Generation Transmission and Distribution, Volume 145, No. 1, pp. 56-64, 1998.

[10] Purushothama, G.K., Narendranath, U.A., and Jenkins, L., "Unit Commitment Using a Stochastic Extended Neighbour Hood Search", IEE Proceedings on Generation Transmission and Distribution, Volume 150, No. 1, pp. 67-72, 2003.

[11] Ongsakul, W.P., and Nit, P., "Unit Commitment by Enhanced Adaptive Lagrangian Relaxation", IEEE Transactions, PS-19, No. 1, pp. 620-628, 2004.

[12] Rajan, C.C.A., Mohan, M.R., and Manivannan, K., "Neural Based Tabu Search Method for Solving Unit Commitment Problem", IEE Proceedings on Generation Transmission and Distribution, Volume 150, No. 4, pp. 469-474, 2003.

[13] Ouyang, Z, and Shahidehpour, S.M., "A Hybrid Artificial Neural Network-Dynamic Programming Approach to Unit Commitment", IEEE Transactions on Power System, Volume 7, No. 2, pp. 236-42, 1992.

[14] Sasaki, H., Fujii, Y., Watanbe, M., Kubokawa, J., Yorino, N., "A Solution Method of Unit Commitment by Artificial Neural Networks", IEEE Transactions on Power Systems, Volume 7, No. 3, August, 1992.

[15] Sasaki, H., Fujii, Y., Watanbe, M., Kubokawa, J., Yorino, N., "A Solution Method Using Neural Network for the Generator Commitment Problem", Electrical Eng in Japan Volume 112, pp. 55-61, 1992.

[16] Yalcinoz, T., Short, M.J., and Cory, B.J., "Application of Neural Networks to Unit Commitment", IEEE Transactions on Power System, pp. 649-654, 1999.

[17] Wong, M.H., Chung, T.S., and Wong, Y.K., "An Evolving Neural Network Approach in Unit Commitment Solution", Microprocessors and Microsystems, Volume 24, pp. 251-262, 2000.

[18] Ouyang, Z., and Shahidehpour, S.M., "A Multi-Stage Intelligent System for Unit Commitment", IEEE Transactions on Power Systems, Volume 7, No. 2, May, 1992.

[19] Huang, S.J., and Huang, C.L., "Application of Genetic-Based Neural Networks to Thermal Unit Commitment", IEEE Transactions on Power Systems, Volume 12, No. 2, May, 1997.

[20] Valle, Y.D., et. al., "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems", IEEE Transactions on Evolutionary Computation, Volume 12, No. 2, pp.171-195, April, 2008.

[21] Mohagheghi, S., Del, Y., Venayagamoorthy, G., and Harley, R., "A Comparison of PSO and Back Propagation for Training RBF Neural Networks for Identification of a Power System with STATCOM", IEEE Proceedings in Swarm Intelligence Symposium, pp. 381-384, June 2005.

[22] Gudise V.G., and Venayagamoorthy, G.K "Comparison of Particle Swarm Optimization and Back Propagation as Training Algorithms for Neural Networks", IEEE Proceedings in Swarm Intelligence Symposium, pp. 110-117, April, 2003.

[23] Feng, J.C., "A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent network Design", IEEE Transactions on System , Man, Cybern, Part-B: Cybern, Volume 34, No. 2, pp. 997-1006, April, 2004.

[24] Grimaldi, I., Grimaccia, E.M., Mussetta, F. M., and Zich, R.E., "PSO as an Effective Learning Algorithm for Neural Network Applications", Proceedings International Conference on Computational Electromagnetic and its Applications, 2004.

[25] Kennedy, J., and Eberhart, R.C., "Particle Swarm Optimization", Proceedings of the IEEE International Conference on Neural Networks, Volume IV, pp. 1942-1948. Perth, Australia, 1995.

[26] Eberhart, R.C., and Kennedy, J., "A New Optimizer using Particle Swarm Theory", Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39-43, Nagoya, Japan, 1995.

[27] Gudise V.G., and Venayagamoorthy, G.K., "Comparison of Particle Swarm Optimization and Backpropagation as Training Algorithms for Neural Networks", Proceedings of the IEEE Swarm Intelligence Symposium, 2003.

[28] Shi, Y.H., and Eberhart, R.C., "A Modified Particle Swarm Optimizer", IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, 1998.

[29] Shi, Y.H., and Eberhart, R.C., "Parameter Selection in Particle Swarm Optimization", 7th Annual Conference on Evolutionary Programming, San Diego, USA, 1998.

[30] Clerc, M., "The Swarm and the Queen, Towards a Deterministic and Adaptive Particle Swarm Optimization", Proceedings of Congress on Evolutionary Computation, pp 1951-1957, Washington DC, 1999.

[31] Bergh, F., Van Den, "An Analysis of Particle Swarm Optimizers", Ph.D. Dissertation, Department of Computer Science, University of Pretoria, Pretoria, South Africa, 2002.

[32] Simi, P., and Swarup, .K.S., "Hopfield Neural Network Approach to the Solution of Economic Dispatch and Unit Commitment" Proceedings of International conference on Intelligent Sensing and Information Processing, pp. 311-316, 2004.

[33] Swarup, K.S., and Yamashiro, S., "A Genetic Algorithm Approach to the Solution of the Basic Unit Commitment Problem", IEE National Convention Conference, pp. 29-34, Japan, October 1997.