

Hybrid Particle Swarm Optimization: with Evolutionary Programming and Shuffled Frog Leaping Algorithm for Long-term Generation Maintenance Scheduling to Enhance the Grid Reliability

G. Giftson Samuel¹ *Member, IEEE* and C. Christober Asir Rajan²

¹Professor, Department of EEE, Sir Issac Newton College of Engineering and Technology, Nagapattinam, India

E-mail: giftsam2k@yahoo.com

²Professor, Department of EEE, Pondicherry Engineering College, Puducherry, India

E-mail: asir_70@pec.edu

Abstract—This paper discuss a Hybrid Particle Swarm Optimization – Genetic Algorithm and Particle Swarm Optimization – Shuffled Frog Leaping Algorithm to Long-term Generation Maintenance Scheduling to Enhance the Reliability of the units. Maintenance scheduling establishes the outage time scheduling of units in a particular time horizon. In a monopolistic power system, maintenance scheduling is being done upon the technical requirements of power plants and preserving the grid reliability. While in power system, technical viewpoints and system reliability are taken into consideration in maintenance scheduling with respect to the economical viewpoint. In this paper present a Hybrid Particle Swarm Optimization – Genetic Algorithm and Particle Swarm Optimization – Shuffled Frog Leaping Algorithm methodology for finding the optimum preventive maintenance scheduling of generating units in power system. The objective function is to maintain the units as earlier as possible. Varies constraints such as spinning reserve, duration of maintenance and maintenance crew are being taken into account. In case study, IEEE test system consist of 24 buses with 32 generating units is used.

Key words—Generation Maintenance Schedule, Optimization, Shuffled frog leaping algorithm, Hybrid Particle Swarm Optimization – Genetic Algorithm and Particle Swarm Optimization – Shuffled Frog Leaping Algorithm

INTRODUCTION

Under the rapid development around the globe, power demand has increased drastically during the past decade. To meet this demand, the development of power system technology has become increasingly important in order to maintain a reliable and economic electric power supply. One major concern of such development is the optimization of power plant maintenance scheduling. Maintenance is aimed at extending the lifetime of power generating facilities, or at least extending the mean time to the next failure for which repair cost may be significant. In addition, an effective maintenance policy can reduce the frequency of service interruptions and the consequences of these interruptions. In other words, having an effective maintenance scheduling is very important for a power system to operate economically and with high reliability.

The focus of this paper was the maintenance decision problem for generation unit system with economic dependency. In the paper, an opportunistic maintenance policy generally applicable to the economic dependency

problem was proposed for developing optimal maintenance schedule. The advances in computer and information technology have created a strong trend to integrate various operation facilities into large-scale system. As a result of this integration, the productivity and efficiency of these systems have been significantly improved. On the other hand, the integration has created a strong functional dependency between the components of the system. Failure of any one of these components could disable the entire system and hence cause significant financial losses and serious safety problems. Effective maintenance program development has become the major challenge and primary concern for today's system managers.

Many maintenance-scheduling methods have been proposed using conventional mathematical programming methods or heuristic techniques. Heuristic approaches provide the most primitive solution based on trial-and-error approaches. These techniques may not generally lead to the global optimal for a complex problem, i.e. the procedure tends to fall into a local minimum if a starting point is not carefully chosen. Heuristic methods were used earlier in solving maintenance scheduling problems for centralized power systems because of their simplicity and flexibility.

Mathematical optimization based techniques such as integer programming (Dopazo and Merrill, 1975), dynamic programming (Zurn and Quintana, 1975; Yamayee et al, 1983) and branch-and-bound (Egan et al, 1976) have been proposed to solve maintenance scheduling problems. For small problems these methods give an exact optimal solution. However, as the size of the problem increases, the size of the solution space increases greatly and hence the running time of these algorithms. These approaches tend to suffer from an excessive computational time with the increase of variables. To overcome this difficulty, modern techniques such as simulated annealing (Cerny, 1985; Kirkpatrick et al, 1983), stochastic evolution (Saab, 1991), genetic algorithms (Goldberg, 1985) and Tabu search (Rajan and Mohan, 2004) have been proposed as alternatives where the problem size precludes traditional techniques. These techniques are completely distinct from classical programming and trial-and-error heuristic methods. The Generic Algorithm method mimics the principles of natural genetics and natural selection to constitute search

and optimization procedures. Simulated annealing mimics the cooling phenomenon of molten metal's to constitute a search procedure. The Generic Algorithm and Simulated Annealing approaches have been reported to solve a range of optimization problems in electrical power systems with encouraging results (Mirinda et al, 1998). Fuzzy optimization techniques have been developed to solve optimal power flow with fuzzy constraints (Xiaohong (Guan and Peter, 1996; Tomsovic, 1992; Miranda et al, 1992), and to schedule manufacturing system with possible breakdowns (Li et al, 1994) The major limitation of these approaches is to consider each generating unit separately in selecting its outage interval, large computational time and complexity in programming.

A little effort has been reported to implement MOPSO for solving power system problems. A fuzzified MOPSO (Wang and Singh, 2007) to solve environmental/economic dispatch problem with heat dispatch and with multiple renewable energy sources. The approach presents a fuzzification mechanism for the selection of global best individual with interpreting the global best as an area, not just as a point. On the other hand, only one local best solution is maintained for each particle. This will degrade the search capability and violates the principle of multiobjective optimization. A modified MOPSO (Kitamura et al, 2005) to optimize an energy management system where the problem is solved in three phases by dividing the original optimization problem into partial problem. However, this approach has severe limitation in the case of strong interaction among the constraints in different subprogram. A MOPSO (Hazra and sinha, 2007) based approach to solve the congestion management problem where the cost and congestion are simultaneously minimized. PSO has been successfully implemented to different power system optimization problem including the economic power dispatch problem with impressive success. (Al-Rashidi et al. 2007) The potential of PSO to handle non smooth and non convex economic power dispatch problem was demonstrated (Selvakumar and Thanushkodi, 2007). However, the problem was formulated as a conventional dispatch problem with the fuel cost as the only objective considered for optimization. Shuffled frog leaping algorithm has been successfully applied to several engineering optimization problems such as unit commitment (M. Eslamian et al. 2009) and job-shop scheduling arrangement (A. Rahimi-Vahed and A. H. Mirzaei 2007).

A novel mechanism (Changyou and Xifan Wang, 2010) for unit maintenance scheduling (UMS) in the deregulated environment, based on the different functions of power producers and the independent system operator (ISO). The proposed scheme aims to achieve a tradeoff between ensuring the producers' benefits and maintaining the system reliability, providing satisfactory maintenance windows and cost-reflective reward/charge to individual producers. Although this can extend anymore, such as the preventing from market power, UMS coordination mechanism and the mechanism of performing the auction sale. A novel concept (Chin Aik

Koay and Srinivasan D 2003) for the spawning and selection mechanism in a hybrid particle swarm algorithm. The results suggest that this hybrid model converges to a better solution faster than the standard PSO algorithm. The hybrid approach proposed here (SPSOES) with spawning and selection mechanism proves to be superior over classical PSO in the cost obtained. Although SPSOES is not as time efficient as standard PSO. A model (Suresh and Kumarappan 2013) for maintenance scheduling (MS) of generators using hybrid improved binary particle swarm optimization (IBPSO) based coordinated deterministic and stochastic approach. Genetic algorithm (GA) operators are introduced in the IBPSO to acquire diversified solutions in the search space. Moreover, the hybrid IBPSO based economic dispatch (ED) has been decomposed as a subproblem in the maintenance model. The authors apply their method to determine the preventive maintenance schedule in a power system. They mention that the method could produce better solutions if some changes and modification are made to the solution procedure.

The proposed algorithm is based on a sequential optimization process of both economic and reliability objectives. The economic purpose is the minimization of total variable operating costs (fuel + O&M and interruptible energy). Other economic objectives have been proposed as maintenance costs, fixed and variable costs, maintenance crew costs, etc. The second optimization run is done minimizing the sum of the differences between the thermal reserve margins of consecutive periods. The reserve margin is calculated dividing the available thermal capacity by the period peak load. This reliability index is the net reserve divided by the gross reserve in period t . The gross reserve in any period is calculated as the difference between the sum of the capacity of all units and the power demand. The net reserve is calculated as the difference between the gross reserve and the power capacity in maintenance. Generally, it is shown that optimal solutions obtained under one reliability criterion are also acceptable in terms of the others. Here, it has been used the net thermal reserve margins leveling between periods in a deterministic way, units availability is modeled derating the maximum unit power by its equivalent forced outage rate.

From the literature review, it has been observed that existed need for evolving simple and effective methods, for obtaining an optimal solution. In this paper an attempt has been made using hybrid PSO-GA and PSO-EP algorithm for meeting the above requirement, which eliminate the drawbacks. In this environment, management of generator and grid is separated, each maximizing its own benefit. Therefore, the principle to draw up the unit maintenance scheduling will be changed significantly. So every generator hopes to put its maintenance on the weeks when market clearing price (MCP) is lowest so that maintenance variable cost descends. The objective function is to sell electricity as much as possible, according to the market clearing price. But the goal of the grid is to maximize the reserve

capacity at every time interval. Depending upon the fitness, profit and reliability index select maintenance scheduling by taking various technical constraints. In the application on IEEE RTS (reliability test system) consist of 24 bus (32 Units) that we can find the optimal solution effectively and these result are compared.

I. PROBLEM FORMULATION

The objective is to find the generation maintenance scheduling, such that minimize total operating cost over the operational planning period and to maximizing the profit, subject to unit maintenance and variety of system constraints.

$$\begin{aligned} \text{Min } F_T = \sum_{t=1}^T \sum_{i=1}^N \{ F_{it}(P_{it}) n_t \} U_{it} + \\ \{ (P_{it} + R_{it}) OMVC n_t \} U_{it} \sum_{t=1}^T \sum_{i=1}^N \{ P_{\max i} OMFC n_t \} \end{aligned} \quad (1)$$

$$\text{Profit} = \sum MCP * P_{it} - F_T \quad (2)$$

$$F_{it}(P_{it}) = A_i + B_i P_{it} + C_i P_{it}^2 \text{ Rs/hr} \quad (3)$$

CONSTRAINTS OF MAINTENANCE SCHEDULING PROBLEM

There are typical constraints for maintenance scheduling problems. Any maintenance timetable must satisfy a given set of constraints. In order to make the maintenance schedule feasible, certain constraints should be fulfilled. Some of basic constraints which should be set up are continuousness maintenance of some unit, maintenance manpower, maintenance window, maintenance duration and so on.

Load Balance

$$\sum_{i=1}^N U_{it} P_{it} = D_t \quad (4)$$

Generator Output Limit

Each unit is designed to work between minimum and maximum power capacity. The following constraint ensures that unit is within its respective rated minimum and maximum capacities.

$$U_{it} P_{i \min} \leq P_{it} \leq U_{it} P_{i \max} \quad (5)$$

Spinning Reserve

Spinning reserve is a safety margin that usually is given as a demand proportion. This indicates that the total capacity of the units running at each interval should not be less than the specified spinning reserve for that interval.

$$\sum_{i=1}^N U_{it} P_{i \max} \geq D_t (1 + r_t \%) \quad (6)$$

Maintenance Resources

$$\sum_{i=1}^N R_i(k) (1 - U_{it}) \leq \alpha_t(k) \quad (7)$$

Maintenance Area

A maximum number of maintenance is imposed in the period t.

$$\sum_{i=1}^u (1 - U_{it}) \leq \beta \quad (8)$$

Crew Constraints

There is limited available manpower in each maintenance area.

$$\sum_{i=0}^v (1 - U_{it}) < M_{it} \quad (9)$$

Fuel Constraints

In some cases thermal units may face fuel shortages. Then requires energy should be purchased from outside.

$$\sum_{t=1}^M F_{it} = Y_i \quad (10)$$

Maintenance Window

The maintenance timetable stated in terms of maintenance variables (S_i). The unit maintenance may not be scheduled before their earliest period or after latest period allowed for maintenance.

$$U_{it} = \begin{cases} 1 & t \leq e_i \text{ or } t \geq l_i + d_i \\ 0 & s_i \leq t \leq s_i + d_i \\ 0, 1 & e_i \leq t \leq l_i \end{cases} \quad (11)$$

One-Time Maintenance

Each unit has an outage for maintenance just once along the time horizon considered.

$$\sum_{t=1}^T s v_{it} = 1 \quad (12)$$

Reliability Indices

For simplicity most of the time, no uncertainty is considered which means that appropriate unit are provided. Nevertheless, unit forced outage rates can be approximately taken into account derating their corresponding capacities.

$$P_{\max i}^+ = (1 - for_i) * U_{it} * P_{\max i} \quad (13)$$

$$\sum_{i=1}^N P_{\max i} * (1 - for_i) - \sum_{i=1}^N P_i(t) \geq \% r_t * d_t \quad (14)$$

$$I(t) = \frac{\sum_{i=1}^N \sum_{t=1}^T P_{it} (1 - U_{it}) (1 - for_i) - D_t}{\sum_{i=1}^N \sum_{t=1}^T P_{it} (1 - for_i) - D_t} \quad (15)$$

In this paper, the focused much attention on maintenance scheduling problems for power systems in order to improve the economic posture of the generation companies. Reducing the total generation cost, including the fuel cost, operation and maintenance cost is one of the main objectives in power system maintenance scheduling.

PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is inspired from the collective behavior exhibited in swarms of social insects. It has turned out to be an effective optimizer in dealing with a broad variety of engineering design problems. In Particle swarm optimization, a swarm is made up of many particles, and each particle represents a potential solution (i.e., individual). A particle has its own position and flight velocity, which keep being adjusted during the optimization process based on the following rules:

$$V_i^{P+1} = \omega * V_i^P + C_1 * rand() * (P_{bi}^{KP} - P_i^{KP}) + C_2 * rand() * (P_{gi}^{KP} - P_i^{KP}) \quad (16)$$

$$P_i^{KP} = P_i^{KP} + V_i^{P+1} \quad (17)$$

where V_{t+1} is the updated particle velocity in the next iteration, V_t is the particle velocity in the current iteration, ω is the inertial dampener which indicates that the impact of the particle's own experience on its next movement, $C_1 * rand$ represents a uniformly distributed number within the interval $[0, c1]$, which reflects how the neighbours of the particle affects its flight, P_{bi}^{KP} is the neighborhood best position, V_i^P is the current position of the particle, $C_2 * rand$ represents a uniformly distributed number within the interval $[0, c2]$, which indicates how the particle trusts the global best position, P_{gi}^{KP} is the global best position, and V_i^{P+1} is the updated position of the particle. Under the guidance of these two updating rules, the particles will be attracted to move toward the best position found thus far. That is, the optimal solutions can be sought out due to this driving force. The major steps involved in Particle Swarm Optimization approach are discussed as follows:

Initialization

The initial particles and velocities of each particle are also selected randomly. The size of the swarm will be $(N_p \times n)$, where N_p is the total number of particles in the swarm and 'n' is the number of stages.

Updating The Velocity

The velocity is updated by considering the current velocity of the particles, the best fitness function value among the particles in the swarm. The velocity of each particle is modified by using equation (16). The value of the weighting factor ω is modified by following equation (18) to enable quick convergence.

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) / iter_{\max} * iter \quad (18)$$

The term $\omega < 1$ is known as the "inertial weight". It is a friction factor chosen between 0 and 1 in order to determine to what extent the particle remains along its original course unaffected by the pull of the other two terms. It is very important to prevent oscillations around the optimal value.

Updating The Position

The position of each particle is updated by adding the updated velocity with current position of the individual in the swarm.

Evolutionary Programming

Evolutionary Programming (EP) can be traced back to early 1950s when Turing discovered a relationship between machine learning and evolution. Later, Bremermann, Box, Friedberg and others put the bases for the evolutionary computation as a tool for machine learning and an optimization technique. Great attention was given to Evolutionary Programming as a powerful tool when Fogal, Burgin, Atmar and others used it to create artificial intelligence to predict the events of a finite state machines (FSM) on the bases of old observation. With advance of the computer performance during 1980s, evolutionary programming was used to solve difficult real world optimization problems. In power system area, Evolutionary Programming has been used to solve a number of problems.

EP is a mutation-based evolutionary algorithm (Cau 2002) and (Nidul Sinha 2003) applied to discrete search spaces. David Fogel extended the initial work of his father Larry Fogel 1962 for applications involving real-parameter optimization problems. Real-parameter Evolutionary Programming is similar in principle to evolution strategy (ES), in that normally distributed mutations are performed in both algorithms. Both algorithms encode mutation strength (or variance of the normal distribution) for each decision variable and a self-adapting rule is used to update the mutation strengths. Several variants of Evolutionary Programming have been suggested.

PROPOSED HYBRID ALGORITHM FOR PARTICLE SWARM OPTIMIZATION (PSO) BASED EVOLUTIONAY PROGRAMMING (EP) FOR MAINTENANCE SCHEDULING

The step by step procedure to compute the global optimal solution is follows:

Step 1: Initialize a population of particles with random positions and velocities on dimensions in the problem space.

Step 2: For each particle, evaluate the desired optimization fitness function in the variables.

Step 3: Compare particles fitness evolution with particles P_{best} . If current value is better then P_{best} , then set P_{best} value equal to the current value, and the P_{best} location equal to the current location in the dimensional space.

Step 4: Compare fitness evaluation with the populations overall previous best. If current value is better than g_{best} , then reset to the current particles array index and value.

Step 5: Change the velocity and position of the particle according to equations (3) and (4) respectively.

Step 6: Loop to step 2 until a criterion is met, usually a sufficiently good fitness or a maximum number of iterations.

Step 7: Initialize the parent vector $p = [p_1, p_2, \dots, p_n]$, $i = 1, 2, \dots, N_p$ such that each element in the vector is determined by $p_j \sim \text{random}(p_{j\min}, p_{j\max})$, $j = 1, 2, \dots, N$, with one generator as dependent generator.

Step 8: Calculate the overall objective function is given in equation (1) using the trail vector p_i and find the minimum of F_{Ti} .

Step 9: Create the offspring trail solution p_i using the following steps.

(a) Calculate the standard deviation

$$\sigma_j = \beta(F_{Tij} / \min(F_{Ti})) (P_{j\max} - P_{j\min})$$

(b) Add a Gaussian random variable to all the state variable of p_i , to get p_i .

Step 10: Select the first N_p individuals from the total $2N_p$ individuals of both p_i & p_i using the following steps for next iteration.

(a) Evaluate $r = (2N_p \text{ random}(0,1) + 1)$

(b) Evaluate each trail vector by $W_{pi} = \sum(W_x)$ Where $x = 1, 2, \dots, N_p$, $i = 1, 2, \dots, 2N_p$ such that $W_x = 1$ if $F_{Tij} / (F_{Tij} + F_{Tir}) < \text{random}(0,1)$, otherwise, $W_x = 0$.

Step 11: Sort the W_{pi} in descending order and the first N_p individuals will survive and are transcribed along with their elements to form the basis of the next generation.

Step 12: The above procedure is repeated from Step 8 until a maximum number of generations N_m is reached.

Step 13: Selection process is done using Evolutionary strategy.

CASE STUDY

A IEEE RTS (reliability test system) is an IEEE 24 bus system with thirty two generating units has been considered as a case studies. A time period of 52 weeks is considered for solving the thirty two units. The proposed methodology was tested for IEEE-RTS is a IEEE 24 bus system with thirty two units.

In this paper maintenance cost models have been developed with and without failure scenarios. Both models include different maintenance cost components which may capture a realistic scenario in a real market environment. In order to demonstrate the solution methodology using the Particle swarm optimization - Genetic Algorithm (PSO-GA) technique for solving Generation Maintenance Scheduling problems, a test system with twelve generating units which must be maintained over a 52 week planning horizon is described in detail here.

Table 1 shows the operation and maintenance cost for IEEE-RTS ie IEEE 24 bus system with thirty two generating units

Table 1 Operation and maintenance cost of 32 units

Units	P_{\max} (MW)	Fixed O & M cost (Rs/KW/yr)	Variable O & M cost (Rs/MWhr)
1 – 5	12	4,50,000	18,000
6 – 9	20	4,05,000	15,000
10 – 13	76	4,00,000	13,500

14 – 19	100	3,50,000	11,250
20 – 23	155	3,15,000	9,500
24 – 29	197	3,10,000	9,000
30	350	2,70,000	7,300
31 & 32	400	2,25,000	5,750

Table 2 shows the generating units data for IEEE-RTS ie IEEE 24 bus system with thirty two generating units.

Table 2 Generating unit data of 32 units

Units	P_{\max} (MW)	Forced Outage Rate (for)	Schedule Maintenance weeks/year	Manpower required per Week
1 – 5	12	0.02	1	10
6 – 9	20	0.1	1	10
10 – 13	76	0.02	3	15
14 – 19	100	0.04	4	15
20 – 23	155	0.04	5	15
24 – 29	197	0.05	6	20
30	350	0.08	8	20
31 & 32	400	0.12	8	20

Table 3 gives data on weekly peak load in percentage of annual peak load.(IEEE RTS data [24])

Table 3 Weekly peak load in percent of annual peak

We ek	Pea k Loa d	We ek	Pea k Loa d	We ek	Pea k Loa d	We ek	Pea k Loa d
1	86.2	14	75.0	27	75.5	40	72.4
2	90.0	15	72.1	28	81.6	41	74.3
3	87.8	16	80.0	29	80.1	42	74.4
4	83.4	17	75.4	30	88.0	43	80.0
5	88.0	18	83.7	31	72.2	44	88.1
6	84.1	19	87.0	32	77.6	45	88.5
7	83.2	20	88.0	33	80.0	46	90.9
8	80.6	21	85.6	34	72.9	47	94.0
9	74.0	22	81.1	35	72.6	48	89.0
10	73.7	23	90.0	36	70.5	49	94.2
11	71.5	24	88.7	37	78.0	50	97.0
12	72.7	25	89.6	38	69.5	51	100
13	70.4	26	86.1	39	72.4	52	95.5

IMPLEMENTATION OF PSO - EP

In this paper, a Particle swarm optimization (PSO) - Evolutionary Programming (EP) based algorithm for solving the Generation Maintenance Scheduling problems will be introduced in which the equality and inequality constraints of the Generation Maintenance Scheduling problems when modifying each particles search point in the Particle swarm optimization algorithm are set. In the initialization process, a set of particles is created in a random order. The structure of a particle for Generation Maintenance Scheduling problems is

composed of a set of elements (i.e. Thermal generations, reserve in each interval). Therefore, particles position at iteration in period of t can be represented by the vector.

To modify the position of each particle, it is necessary to calculate the velocity of each particle. In this position updating process, the value of parameters such as ω , C_1 , C_2 and C_3 Should be determined in advance. In this thesis, the inertial weight is defined as Equation 18. The position of each particle is based on improving fitness or achievement of objective function. Thus, each particle keeps the previous best position and corresponding fitness until the next velocity which leads to new best position and achievement.

At first, this approach is applied to the test system to obtain the acceleration constant C_1 , C_2 and C_3 . Then two different test systems were tested to verify the feasibility solution of the proposed method to solve Generation Maintenance Scheduling problem. The value of ω_{\max} , ω_{\min} and $iter_{\max}$ are 1.0, 0.1 and 100 respectively. The other parameter such as C_1 , C_2 and C_3 are selected through the evaluation of the output, after many runs on the test system.

To solve an optimization problem using Particle Swarm Optimization based Evolutionary Programming (PSO – EP), first the possible solutions of the problem have to be coded in chromosomes. Next a fitness function to compare the chromosomes has to be defined. The period of maintenance scheduling is usually one year and is divided into T stages. When a stage is one week, T is equal to 52. In solving the generation maintenance problem, the main variables to be identified are maintenance states of the generating units. The schedule for unit i could be represented by a string of zeros and ones, U_{ij} where one means the unit i is under maintenance in the stage t . We take the maintenance schedule corresponding to an individual generating unit as a gene and build the chromosome from these genes. Therefore a single chromosome will completely describe the maintenance schedules for power generating units. Next the last generation individuals are taken as input for Evolutionary Programming, and then offspring will be generated and best individuals are taken for transcribed along with their elements. Finally the selection process will be done using Evolutionary Programming.

In this Evolutionary Programming a set of experiments for the twelve generator units was carried out. Evolutionary Programming search technique moves from one solution to another using a probabilistic search method. However, the new solution may render infeasible. Therefore, using Evolutionary Programming alone may take a long time to reach the optimal solution or it may get trapped in a local optimum. So, the Hill-Climbing technique is used in conjunction with Evolutionary Programming to find a feasible solution in the neighborhood of the new infeasible solution. The Evolutionary Programming search ability and the feasibility watch of the Hill-Climbing motivate the sequential solution of the Generation Maintenance Scheduling problem. Selecting the individuals, which survive to the next generation, is based on the overall objective function.

To solve for Generation Maintenance Scheduling problem, the proposed hybrid methodology was developed Particle Swarm Optimization based Evolutionary Programming (PSO – EP) program has been carried out on a Pentium IV 2-GHz PC with a 512 Mbyte RAM (in MATLAB 7.3). The software provides interactive approach in dealing with the various data input required for solving the Generation Maintenance Scheduling with constraints which should be set up are the continuity of maintenance activity, specific time interval for maintenance of some generating units, maximum and minimum capacity of each generating unit, minimum net reserve.

The annual peak load for the thirty two generator test system is 2850 MW. Each unit must be maintained (without interruption) for a given duration within an allowed period. The allowed period for each generator is the result of a technical assessment and the experience of the maintenance personnel, which ensures adequate maintenance period.

Figure 1 shows performance of the hybrid Particle Swarm Optimization based Evolutionary Programming (PSO – EP) over 1000 generations when maintenance scheduling of the generating units formulated based on its desired objective function and figure, the fitness of the population are illustrated. For the maximization problem fitness function is same as the objective function.

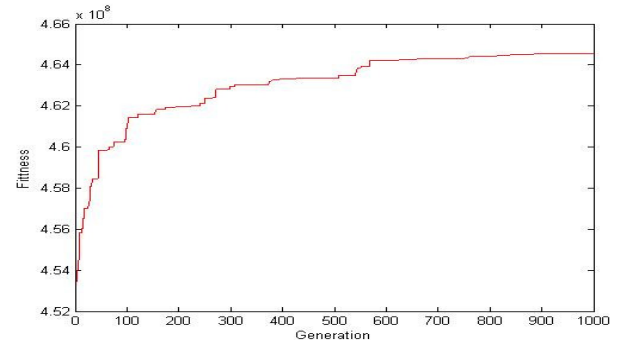


Figure 1 Performance of object function for 32 units

Figure 2. Shows the during the week 8 maximum generator maintenance is carried out.

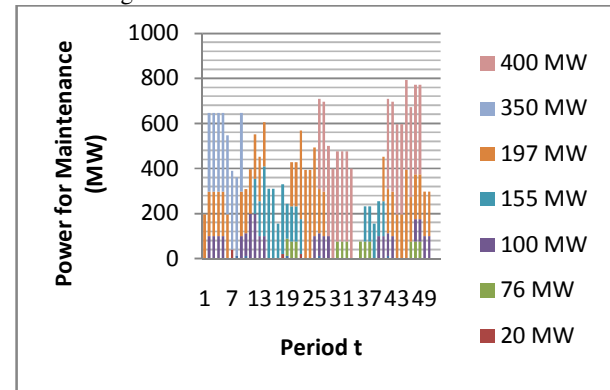


Figure 3. shows maintenance schedule of generating units and the profit is maximum during the week 51.

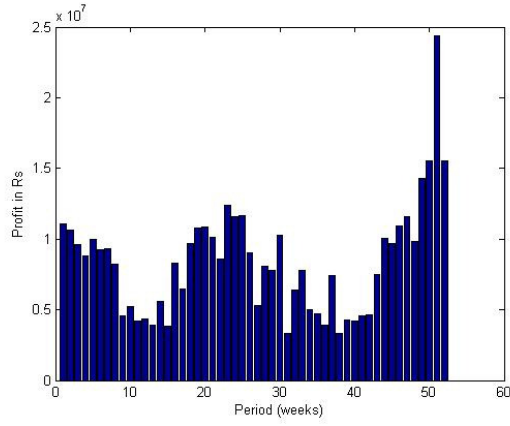


Figure 3 Scheduling of objective function for 32 Units

To measure the degree of security throughout the weeks of the year, the reliability index is defined for period t . The reliability index is shown in Figure 4 for thirty two generating units. This reliability index is the net reserve divided by the gross reserve in period t . The gross reserve in any period is calculated as the difference between the sum of the capacity of all units and the power demand. The net reserve is calculated as the difference between the gross reserve and the power capacity in maintenance.

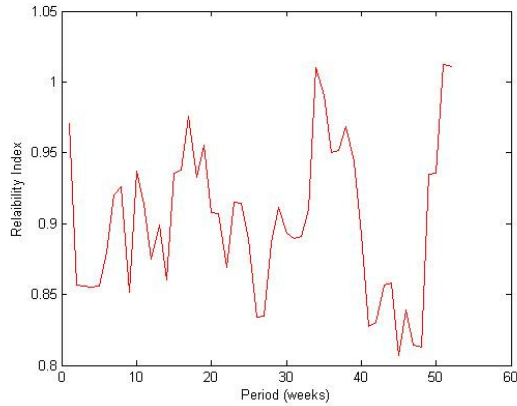


Figure 4. Reliability index for objective function for 32 units

SHUFFLED FROG LEAPING ALGORITHM

The Shuffled frog leaping algorithm is a meta-heuristic optimization method which is based on observing, imitating, and modeling the behavior of a group of frogs when searching for the location that has the maximum amount of available food [19]. Shuffled frog leaping algorithm, originally developed by Eusuff and Lansey in 2003, can be used to solve many complex optimization problems, which are nonlinear, non differentiable, and multi-modal [20]. Shuffled frog leaping algorithm has been successfully applied to several engineering optimization problems such as water resource distribution [21], bridge deck repairs [22], job-shop scheduling arrangement [23], and traveling salesman problem (TSP) [24]. The most distinguished

benefit of Shuffled frog leaping algorithm is its fast convergence speed [25]. The Shuffled frog leaping algorithm combines the benefits of the both the genetic-based memetic algorithm (MA) and the social behavior-based Particle Swarm Optimization algorithm.

Shuffled frog leaping algorithm is a population based random search algorithm inspired by nature memetics. In the Shuffled Frog Leaping algorithm, a population of possible solution defined by a group of frogs that is partitioned into several communities referred to as memeplexes. Each frog in the memeplexes is performing a local search. Within each memeplex, the individual frog's behavior can be influenced by behaviors of other frogs, and it will evolve through a process of memetic evolution. After a certain number of memetics evolution steps, the memeplexes are forced to mix together and new memeplexes are formed through a shuffling process. The local search and the shuffling processes continue until convergence criteria are satisfied. The flowchart of Shuffled frog leaping algorithm is illustrated in various steps as follows:

Step 1: The Shuffled frog leaping algorithm involves a population 'P' of possible solution, defined by a group of virtual frogs(n).

Step 2: Frogs are sorted in descending order according to their fitness and then partitioned into subsets called as memeplexes (m).

Step 3: Frogs i is expressed as $X_i = (X_{i1}, X_{i2}, \dots, X_{is})$ where S represents number of variables.

Step 4: Within each memeplex, the frog with worst and best fitness are identified as X_w and X_b .

Step 5: Frog with global best fitness is identified as X_g .

Step 6: The frog with worst fitness is improved according to the following equation.

$$D_i = \text{rand}() \cdot (X_b - X_w) \quad (19)$$

$$X_{\text{neww}} = X_{\text{oldw}} + D_i \quad (-D_{\text{max}} \leq D_i \leq D_{\text{max}}) \quad (20)$$

Where rand is a random number in the range of [0,1];

D_i is the frog leaping step size of the i^{th} frog and D_{max} is the maximum step allowed change in a frog's position. If the fitness value of new X_w is better than the current one, X_w will be accepted. If it isn't improved, then the calculated (19) and (20) are repeated with X_b replaced by X_g . If no improvement becomes possible in the case, a new X_w will be generated randomly. Repeat the update operation for a specific number of iterations. Therefore, Shuffled frog leaping algorithm simultaneously performs an independent local search in each memeplex using a process similar to the Particle Swarm Optimization algorithm. The flowchart of local search of Shuffled frog leaping algorithm is illustrated in Figure 5.

After a predefined number of memetic evolutionary steps within each memeplex, the solutions of evolved memeplexes are replaced into new population. This is called the shuffling process. The shuffling process promotes a global information exchange among the frogs. Then, the population is sorted in order of decreasing performance value and updates the population best frog's

position, repartition the frog group into memeplexes, and progress the evolution within each memeplex until the conversion criteria are satisfied. Usually, the convergence criteria can be defined as follows :

The relative change in the fitness of the global frog within a number of consecutive shuffling iterations is less than a pre-specified tolerance.

The maximum predefined number of shuffling iteration has been obtained

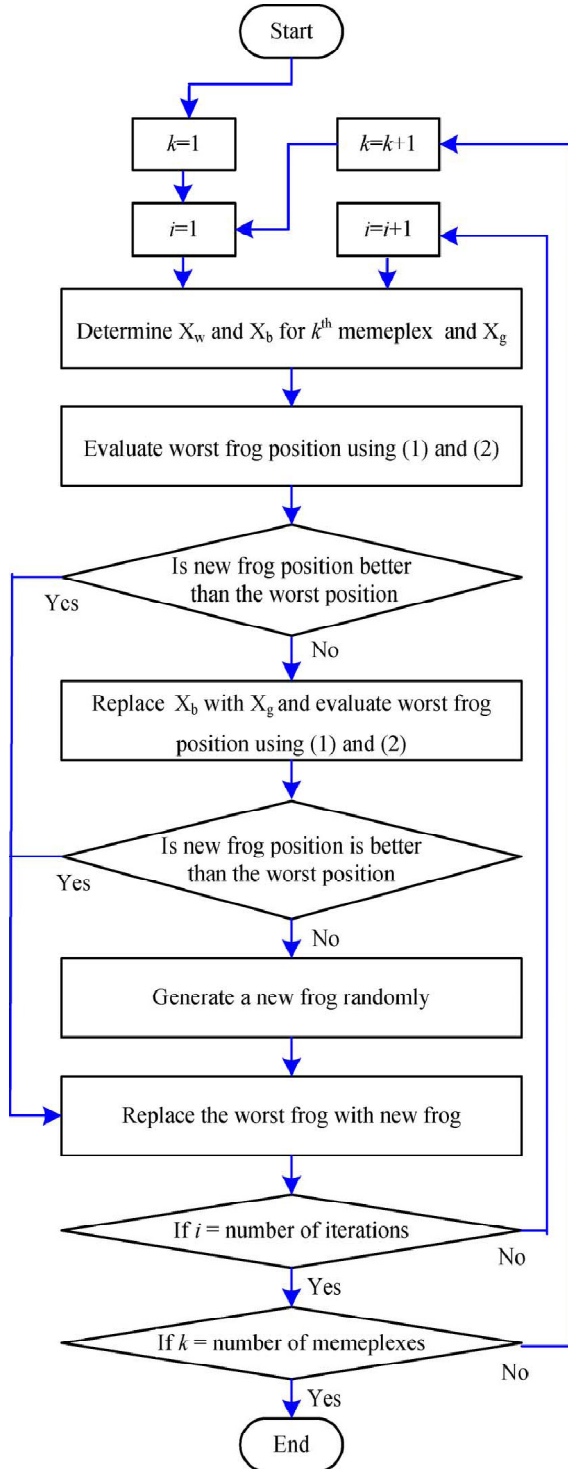


Figure 5 Flowchart of local search

PROPOSED HYBRID ALGORITHM FOR PSO & SFLA FOR MAINTENANCE SCHEDULING

The step by step procedure to compute the global optimal solution is follows:

Step 1: Initialize a population of particles with random positions and velocities on dimensions in the problem space.

Step 2: For each particle, evaluate the desired optimization fitness function in the variables.

Step 3: Compare particles fitness evolution with particles P_{best} . If current value is better then P_{best} , then set P_{best} value equal to the current value, and the P_{best} location equal to the current location in the dimensional space.

Step 4: Compare fitness evaluation with the populations overall previous best. If current value is better than g_{best} , then reset to the current particles array index and value.

Step 5: Change the velocity and position of the particle according to Equations (17) and (18) respectively.

Step 6: Loop to step 2 until a criterion is met, usually a sufficiently good fitness or a maximum number of iterations.

Step 7: Initialize the population 'P' of possible solution, defined by a group of virtual frogs (n).

Step 8: Frogs are sorted in descending order according to their fitness and then partitioned into subsets called as memeplexes (m).

Step 9: Frogs i is expressed as $X_i = (X_{i1}, X_{i2}, \dots, X_{is})$ where S represents number of variables.

Step 10: Within each memeplex, the frog with worst and best fitness are identified as X_w and X_b .

Step 11: Frog with globe best fitness is identified as X_g .

Step 12: The frog with worst fitness is improved according to the Equations (19) and (20).

IMPLEMENTATION OF PSO - SFLA

The total operating cost of the Generation Maintenance Scheduling problem is expressed as the sum of fuel costs, operation and maintenance variable costs (OMVC), operation and maintenance fixed costs (OMFC) of the generating units. The fuel cost is the major component of the operating cost, which is normally modeled by a quadratic input/output curve. The Generation Maintenance Scheduling problem can be formulated as an integer programming problem by using integer variable variables to represent the period in which the maintenance of each unit starts. The variables are bounded by maintenance window constraints. However, for clarity the problem is first formulated using binary variables which indicate the start of maintenance of each unit at each time. A reliability criterion is considered to formulate the Generation Maintenance Scheduling problem. The leveling of the reserve generation over the planning period can be used as a reliability criterion. The net reserve of the system during any period t is the total installed capacity minus the peak load forecast for the period and the reserve loss due to the pre-scheduling outage. The reserve can be leveled by maximizing net reserve of the system during any period.

To solve for Generation Maintenance Scheduling problem, the proposed hybrid methodology was developed Particle Swarm Optimization based Shuffled frog leaping Algorithm (PSO – SFLA) program has been carried out on a Pentium IV 2-GHz PC with a 512 Mbyte RAM (in MATLAB 7.3). The software provides interactive approach in dealing with the various data input required for solving the Generation Maintenance Scheduling with constraints which should be set up are the continuity of maintenance activity, specific time interval for maintenance of some generating units, maximum and minimum capacity of each generating unit, minimum net reserve.

Figure 6 shows performance of the hybrid PSO-SFLA over 1000 generations when maintenance scheduling of the generating units formulated based on its desired objective function and figure, the fitness of the population are illustrated. For the maximization problem fitness function is same as the objective function.

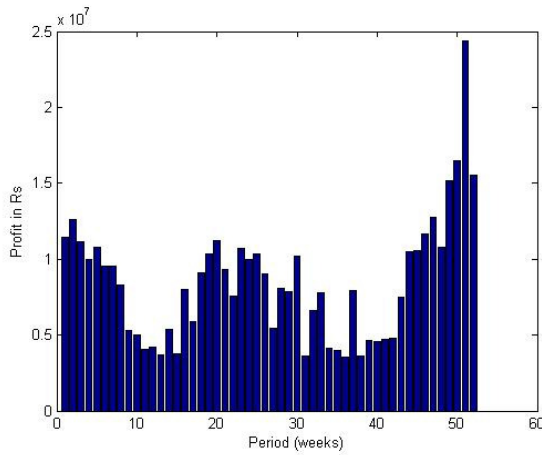


Figure 6 Scheduling of objective function for 32 units

Figure 7 show the week 8 maximum generator maintenance is carried out. Figure 8 shows maintenance schedule of generating units and the profit is maximum during the week 51.

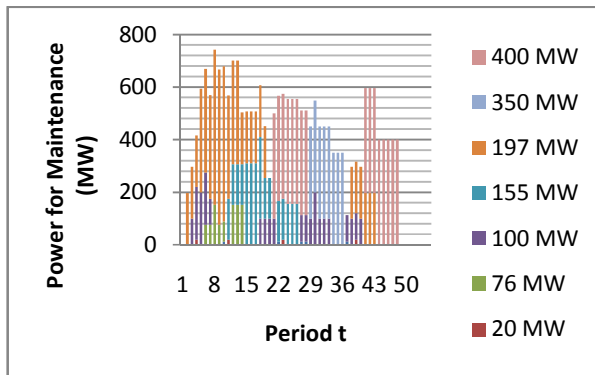


Fig. 7 – Maintenance Scheduling of Objective function for 32 Generator Units

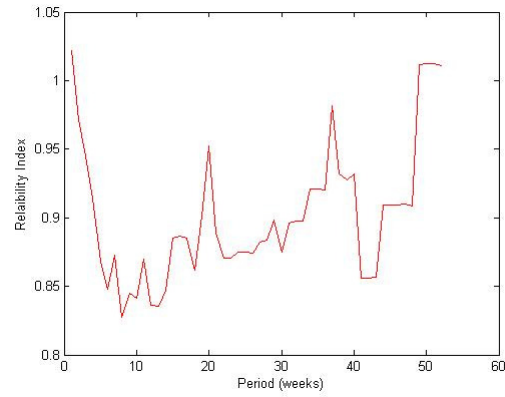


Figure 8 Reliability index for objective function for 32 units

To measure the degree of security throughout the weeks of the year, the reliability index is defined for period t . The reliability index is shown in Figure 8 for thirty two generating units. This reliability index is the net reserve divided by the gross reserve in period t . The gross reserve in any period is calculated as the difference between the sum of the capacity of all units and the power demand. The net reserve is calculated as the difference between the gross reserve and the power capacity in maintenance

NUMERICAL RESULTS AND DISCUSSIONS

A IEEE RTS (reliability test system) – 24 bus system with thirty two generating units has been considered for this work. Fuel cost function of each generator is estimated into quadratic form.

A time period of 52 weeks is considered for solving this maintenance problem for thirty two units. Generation maintenance scheduling with reserve margin is solved for thirty two generating unit by the three hybrid algorithms which gives total cost, computation timing and reliability index.

Two proposed hybrid algorithms were applied and compared with other conventional method and it is shown in Table 4. The comparison of the total cost and central processing unit (CPU) time are as shown in Table 5. The comparison of the total profit with reserve margin for all weeks is as shown in Table 6. The comparison of reliability index is shown in Figure 7 for thirty two generating units. Here, total cost computed by using hybrid Particle Swarm Optimization based Evolutionary Programming (PSO-EP) is higher than Particle Swarm Optimization based Shuffled Frog Leaping Algorithm (PSO-SFLA), but computation time is greater than Particle Swarm Optimization based Shuffled frog leaping Algorithm (PSO-SFLA). When comparing the profit Particle Swarm Optimization based Shuffled Frog Leaping Algorithm (PSO-SFLA) is higher than Particle Swarm Optimization based Evolutionary Programming (PSO-EP). When comparing the reliability index for reserve, the hybrid Particle Swarm Optimization based Shuffled Frog Leaping Algorithm

(PSO-SFLA) has maximum value in week one and hybrid Particle Swarm Optimization based Evolutionary Programming (PSO-EP) have minimum value in week thirty one. When compared with average reserve hybrid Particle Swarm Optimization based Shuffled Frog Leaping Algorithm (PSO-SFLA) has higher value than Particle Swarm Optimization based Evolutionary Programming (PSO-EP).

Three proposed hybrid algorithms were applied and compared with other conventional method and it is shown in Table 4. The comparison of the total cost and central processing unit (CPU) time are as shown in Table 5. The comparison of the total profit with reserve margin for all weeks is as shown in Table 6. The comparison of reliability index is shown in Figure 7 for thirty two generating units. Here, total cost computed by using hybrid Particle Swarm Optimization based Genetic Algorithm (PSO-GA) is higher than Particle Swarm Optimization based Shuffled Frog Leaping Algorithm (PSO-SFLA), but computation time is less than Particle Swarm Optimization based Genetic Algorithm (PSO-GA) and greater than Particle Swarm Optimization based Shuffled frog leaping Algorithm (PSO-SFLA). When comparing the profit Particle Swarm Optimization based Shuffled Frog Leaping Algorithm (PSO-SFLA) is higher than Particle Swarm Optimization based Genetic Algorithm (PSO-GA). When comparing the reliability index for reserve, the hybrid Particle Swarm Optimization based Shuffled Frog Leaping Algorithm (PSO-SFLA) has maximum value in week one and hybrid Particle Swarm Optimization based Genetic Algorithm (PSO-GA) have minimum value in week. When compared with average reserve hybrid Particle Swarm Optimization based Shuffled Frog Leaping Algorithm (PSO-SFLA) has higher value than Particle Swarm Optimization based Genetic Algorithm (PSO-GA).

Table 4 Comparison of generation maintenance schedule of units for 32 generator units

Week	Generating units scheduled for Maintenance	
	PSO-EP	PSO-SFLA
1	27	-
2	16, 27, 30	25
3	16, 27, 30	18, 25
4	16, 27, 30	6, 17, 18, 25
5	16, 27, 30	17, 18, 24, 25
6	27, 30	10, 17, 18 24, 25
7	7, 9, 30	10, 17, 24, 25
8	5, 30	10, 12, 24 26, 29
9	17, 28, 30	12, 24, 26 29
10	1, 17, 28	3, 12, 24 26, 29
11	17, 18, 28	8, 23, 26 29
12	17, 18, 23, 28	11, 13, 23, 26,29
13	18, 23, 28	11, 13, 23, 26,29
14	18, 21, 23, 28	11, 13, 20, 23,28
15	21, 23	20, 23, 28
16	21, 23	20, 22, 28
17	21	20, 22, 28
18	6, 20, 21	14, 20, 22, 28
19	2, 12, 20	14, 22, 28

20	12, 20, 26	14, 22
21	12, 20, 26	14, 31
22	8, 20, 24, 26	4, 21, 31
23	24, 26	7, 21, 31
24	24, 26	21, 31
25	15, 24, 26	21, 31
26	4, 15, 24, 31	21, 31
27	15, 24, 31	2, 15, 31
28	15, 31	5, 15, 31
29	31	15, 30
30	10, 31	15, 16, 30
31	10, 31	16, 30
32	10, 31	16, 30
33	31	16, 30
34	-	30
35	11	30
36	11, 22	30
37	11, 22	1, 19
38	22	19, 27
39	14, 22	9, 19, 27
40	14, 22, 29	19, 27
41	3, 14, 29, 32	27, 32
42	14, 29, 32	27, 32
43	29, 32	27, 32
44	29, 32	32
45	25, 29, 32	32
46	13, 25, 32	32
47	13, 19, 25, 32	32
48	13, 19, 25, 32	32
49	19, 25	-
50	19, 25	-
51	-	-
52	-	-

Table 5 Comparison of cost and CPU time for 32 units

System	Method	Total Cost (pu)	CPU Time (Sec)
32 Units	DP	1.0535	918
	LR	1.0687	850
	PSO	1.0387	417
	EP	1.0392	422
	PSO-EP	1.0385	410
	PSO - SFLA	1.0381	407

Table 6 Comparison of profit/cost for 32 generator units in each week

Week	Generating units scheduled for Maintenance		Week	Generating units scheduled for Maintenance	
	PSO-EP (in million `)	PSO-SFLA (in million Rs)		PSO-EP (in million `)	PSO-SFLA (in million Rs)
1	1.124	1.139	27	0.534	0.556
2	1.039	1.264	28	0.815	0.806
3	0.955	1.111	29	0.787	0.778
4	0.871	1.000	30	1.039	1.028
5	1.011	1.083	31	0.337	0.361
6	0.955	0.944	32	0.646	0.667

7	0.969	0.944	33	0.758	0.778
8	0.815	0.833	34	0.506	0.403
9	0.478	0.528	35	0.449	0.389
10	0.520	0.500	36	0.393	0.361
11	0.421	0.403	37	0.758	0.806
12	0.449	0.417	38	0.337	0.361
13	0.393	0.361	39	0.421	0.472
14	0.562	0.528	40	0.421	0.458
15	0.365	0.361	41	0.449	0.472
16	0.843	0.806	42	0.463	0.486
17	0.646	0.583	43	0.758	0.750
18	0.955	0.917	44	1.011	1.028
19	1.067	1.028	45	0.955	1.056
20	1.070	1.111	46	1.096	1.167
21	1.011	0.931	47	1.152	1.278
22	0.857	0.750	48	0.983	1.083
23	1.236	1.069	49	1.433	1.528
24	1.152	1.000	50	1.545	1.639
25	1.152	1.028	51	2.444	2.444
26	0.899	0.889	52	1.573	1.556
Total in Million`				43.879	44.236

Table 7 Comparison of reliability index for objective function for 32 units

Method	Max Value	Week	Min Value	Week	Average Value
PSO – EP	0.958	8 & 18	0.79	31	0.958
PSO - SFLA	1.00	1	0.825	8	0.925

As a result of maximum profit and better reserve margin, hybrid Particle Swarm Optimization based Shuffled Frog Leaping Algorithm (PSO-SFLA) is the best of achieving objective function among the three proposed algorithms in all IEEE (RTS) systems.

II. CONCLUSIONS

This paper shows a new approach for solving the generation maintenance scheduling problem based on modified Shuffled frog leaping algorithm and the optimum maintenance scheduling over the planning period was obtained. The algorithm has been tested on thirty two generating unit systems. The proposed method has been compared with the results of other method such as Lagrangian Relaxation, Dynamic Programming, Evolutionary Programming, Particle Swarm Optimization and Shuffled frog leaping Algorithm and hybrid Particle Swarm Optimization based Evolutionary Programming (PSO-EP) and Particle Swarm Optimization based Shuffled Frog Leaping Algorithm (PSO – SFLA) give an idea regarding how generator schedule and reserve should be maintained to maximize profit reduce the computation timing.

To conclude, hybrid particle swarm optimization based Shuffled Frog Leaping Algorithm (PSO-SFLA) gives the best solution quality, robust, cost-effective, reserve margin and consumes minimum computation

time for generation maintenance scheduling of thermal units.

III. APPENDIX

A_i, B_i, C_i – the cost function parameters of unit I
(Rs/MW²hr, Rs/MWhr, Rs/hr)

$F_{it}(P_{it})$ – production cost of unit I at a time t (Rs/hr)

P_{it} – output power from unit i at time t (MW)

PD_t – system peak demand at hour t (MW)

N – Number of available generating units.

R_{it} – reserve contribution of unit i at time t.

n_t – number of units

U_{it} – commitment state of unit i at time t (on = 1, off = 0)

OMVC – operation and maintenance variable cost

OMFC – operation and maintenance fixed cost

T_s & T_e – Starting and ending stage of the time interval for j^{th} unit

$I(t)$ - Reliability index of grid in period t.

$\alpha_t(k)$ - k^{th} maintenances resource at the t^{th} period.

β - Maximum number of maintenance generator in the same area.

d_i - Maintenance duration of the i^{th} generator

s_i - Maintenance starting period of the i^{th} generator.

IV. REFERENCES

1. Al-Rashidi MR, El-Hawary (2007). Hybrid particle swarm optimization approach for solving the discrete OPF problem considering the value loading effect. IEEE Trans. Power system. 22(4): 2030-2038.
2. Cau TDH, Kaye RJ. Evolutionary Optimization Method for multi storage hydrothermal scheduling. IEE Proc. – Gener. Transm. Distrib.. 149: 152-156.
3. Cerny V (1985). Thermo Dynamic Approach to the Travelling Salesman Problem: An Efficient Simulation Algorithm. Journal of Optimization Theory and Applications. 45: 41 -52.
4. Changyou and Xifan Wang (2010). A Competitive Mechanism of Unit Maintenance Scheduling in a Deregulated Environment. IEEE Transactions on Power Systems. 25(1): 351-359.
5. Chin Aik Koay and Srinivasan D (2003). Particle swarm optimization-based approach for generator maintenance scheduling. International Conference on Swarm Intelligence Symposium. 1: 167-173.
6. Dopazo JF, Merrill HM (1975). Optimal Generator Maintenance Scheduling using Integer Programming. IEEE Transactions on Power Apparatus and Systems. 94: 1537-1545.
7. Egan GT, Dillon TS, Morsztyn K. An Experimental Method of Determination of Optimal Maintenance Schedules in Power Systems using the Branch-and-Bound Technique. IEEE Transactions on Systems. Man and Cybernetics. 6: 538-547.
8. Eslamian M, Hosseinian MH, Vahidi B (2009). Bacterial foraging-based solution to the unit-commitment problem. IEEE Trans. Power Syst.. 24: 1478-1488.
9. Hazra J and sinha AK (2007). Congestion management using multiobjective particle swarm optimization. IEEE Trans. Power systems. 22(4): 1726-1734.

10. Kirkpatrick S, Gelatt CD, Vecchi M (1983). Optimization by Simulated Annealing. *Science*. 220: 671 -680.
11. Kitamura S, Mori K, Shindo S, Izui Y and Ozaki Y (2005). Multiobjective energy management system using modified MOPSO. *IEEEconf. System man cybern.* 4: 3497-3503.
12. Li Y, Luh PB, Guan X (1994). Fuzzy Optimization-Based Scheduling of Identical Machines with Possible Breakdown. *Proc of IEEE International conference on Robotics*. San Diego. pp 3347-3452.
13. Miranda V, Saraiva IJ (1992). Fuzzy Modeling of Power System Optimal Load flow. *IEEE Trans Power Systems*. 7: 843-849.
14. Mirinda V, Srinivasa D, Proenca LM (1998). Evolutionary Computation in Power Systems, *Electrical Power and Energy*. 20: 89-90.
15. Mirinda V, Srinivasa D, Proenca LM (1998). Evolutionary Computation in Power Systems, *Electrical Power and Energy*. 20: 89-90.
16. Nidul Sinha R, Chakrabarti, Chattopadhyay PK (2003). Fast Evolutionary Programming Techniques for Short-Term Hydrothermal Scheduling. *IEEE Transactions on Power Systems*. 18: 214-220.
17. Rahimi-Vahed A and Mirzaei AH (2007). Solving a bi-criteria permutation flow-shop problem using shuffled frog-leaping algorithm. *Soft Computing*. New York: Springer-Verlag.
18. Rajan, Mohan.M.R (2004). An Evolutionary Programming Based Tabu Search for solving the Unit commitment Problem. *IEEE Trans on Power System*. 19: 577-589.
19. Saab YG, Rao VB (1991). Combinatorial Optimization by Stochastic Evolution. *IEEE Transactions on Computer-Aided Design*. 10: 525-535.
20. Selvakumar AI and Thanuskodi (2007). A new particle swarm optimization solution to nonconvex economic dispatch problem. *IEEE Trans. Power systems*. 22(1): 42-51.
21. Suresh K and Kumarappan N (2013). Hybrid improved binary particle swarm optimization approach for generation maintenance scheduling problem. *Swarm and Evolutionary Computation*. 9: 69-89.
22. The IEEE Reliability Test System – 1996. *IEEE Transaction on Power Systems*. 14: 1010-1020.
23. Tomsovic Y (1992). A Fuzzy Linear Programming Approach to the Reactive Power/Voltage Control Problem. *IEEE Trans on Power System*. 7: 287-293.
24. Wang Lingfeng and Singh Chanan (2007). Environmental/economic power dispatch using fuzzified multi objective particle swarm optimization algorithm. *Electric power systems*. 77: 1654-1664.
25. Xiaohong Guan, Peter B.Luh (1996). Power System Scheduling with Fuzzy Reserve Requirements. *IEEE Trans on Power System*. 11: 864-869.
26. Yamayee ZA, Sidenblad K, Yoshimura M (1983). A Computationally Efficient Optimal Maintenance Scheduling Method. *IEEE Transactions on Power Apparatus and Systems*. 102: 330-338.
27. Zhang X, Hu X, Cui G, Wang Y, Niu Y (2008). An improved shuffled frog leaping algorithm with cognitive behavior. *Proc. 7th World Congr. Intelligent Control and Automation*.
28. Zurn HH, Quintana VH (1975). Generator Maintenance Scheduling via Successive Approximations Dynamic Programming. *IEEE Transactions on Power Apparatus and Systems*. 94: 665-671.