

A Bacterial Foraging PSO – DE Algorithm for Solving Reserve Constrained Dynamic Economic Dispatch Problem

K.Vaisakh

Department of Electrical
Engineering
Andhra University
Visakhapatnam, AP, India
vaisakh_k@yahoo.co.in

P.Praveena

Department of Electrical
Engineering
Andhra University
Visakhapatnam, AP, India
nambaripraveena@gmail.com

S.Rama Mohana Rao

Department of Electrical
Engineering
Andhra University
Visakhapatnam, AP, India
ramu_sanchana@yahoo.com

Abstract--This paper introduces a solution to Dynamic Economic Dispatch (DED) problem using a hybrid optimization methodology bacterial foraging and PSO-DE (BPSO-DE) by integrating Bacterial foraging optimization Algorithm (BFOA), Particle Swarm Optimization (PSO) and Differential Evolution (DE). In the proposed method BFOA performs local search and global search for entire search space is accomplished through PSO-DE operators in this way they move to reach the global optimization. The BFOA also takes care of the constraints such as ramp-rate limits, valve-point loading effects, system load demand, prohibited operating zones, power losses and spinning reserve capacity. A ten unit test system is considered to show the effectiveness of the proposed method over other existing methods.

Index Terms -- Bacterial Foraging, Particle Swarm Optimization, Differential Evolution

I. NOMENCLATURE

D	: Dimension of search space.
N_b	: The number of bacteria in the population.
N_C	: The number of chemo-tactic steps.
N_{re}	: The number of reproduction steps.
N_{ed}	: The number of elimination-dispersal events.
N_S	: Swimming length N_S is the maximum number of steps taken by each bacterium when it moves from low nutrient area to high nutrient area.
P_{ed}	: Probability of elimination and dispersal
CS	: The size of the step taken in the random direction specified by the tumble.
w	: The inertia weight.
$P(t, j, k, l, m)$	: Position vector of j^{th} unit in t^{th} hour, k^{th} bacterial population, l^{th} reproduction and m^{th} elimination-dispersal, where $k = 1, 2, \dots, N_b$, $l = 1, 2, \dots, N_{re}$

	and $m = 1, 2, \dots, N_{ed}$
$V(t, j, k)$	: Velocity vector of the j^{th} unit in t^{th} hour and k^{th} bacterial population.
$pbest$	: The best previous position of k^{th} particles.
$gbest$	: The index of the best particle among all particles in the group.
sf	: The scale factor
CR	: The Cross over probability
C_1, C_2	: Acceleration constants
rand, Rand	: Random value in the range [0,1]
$J(k, l, m)$	: Cost value of k^{th} bacterial population, l^{th} reproduction and m^{th} elimination-dispersal
$JF(k, l, m)$	: Fitness value of k^{th} bacterial population, l^{th} reproduction and m^{th} elimination-dispersal
r_2, r_3	: Parameter vectors chosen randomly from the current population.

II. INTRODUCTION

DED problem determines the optimal generation schedule of on-line generators, so as to meet the predicted load demand over a time horizon satisfying dynamic constraint such as ramp-rate limits. The goal of DED problem is to minimize operating cost under various system and operational constraints. The ramp-rate limits must be properly modeled in production simulation as the operational decision at an hour t may affect the operational decision at a later hour. So DED is a most difficult dynamic optimization problem.

The literature addresses DED problem with convex cost functions [1]–[4], though in reality, large steam turbines generally have a number of fuel admission valves which are opened sequentially in order to increase the production. When the valve is opened, the throttling losses increases rapidly as a

result of which, the incremental heat rate rises suddenly. The valve-point effects and prohibited operating regions contribute non-convexity in the fuel cost function of the generating units [5]-[9]. The generators operating in prohibited operating region experience amplification of vibrations in their shaft bearing [10], [11] due to physical operation limits such as faults in the machines themselves or the associated auxiliaries, such as boiler, feed pumps, etc. Traditional methods like gradient projection method [1], Lagrangian relaxation [12], and dynamic programming when used to solve DED problem often use more approximations to limit the complexity of the problem and take longer solution time. Hence these methods suffer from solution stagnation problem leading them to a less desirable performance. Genetic Algorithm (GA) [5], [13], evolutionary programming (EP) [6], [14]-[16], simulated annealing (SA) [17], [18], particle swarm optimization (PSO) [7] and differential evolution (DE) [19], [20] may prove to be very effective in solving nonlinear ED problems without any restriction on the shape of the cost curves. They often provide fast and reasonable solutions. EP seems to be a good method to solve optimization problems, when applied to problems consisting of more number of local optima the solutions obtained from EP method when combined with sequential quadratic programming (SQP) is just near global optimum[21]-[24]. Recently, a variable scaling hybrid differential evolution (VSHDE) and hybrid differential evolution (HDE) [25] methods are used to solve the DED problem.

The Bacterial Foraging Optimization Algorithm (BFOA) is currently gaining popularity in the community of researchers, for its effectiveness in solving certain difficult real-world optimization problems. BFOA is based on the foraging strategies of Ecoli bacterium cells and was proposed by Prof. K. M. Passino in 2001 [26]. PSO and DE are excellent heuristics like other evolutionary algorithms [29]. Practical experiences suggest that they reach stagnation after certain number of generations as the population is not converged locally, so they will stop proceeding towards global optimal solutions. The stochastic search methods are proven in reaching global solutions for certain difficult real world optimization problems [30]-[36]. Hence this article comes up with a hybrid approach involving PSO-DE and BFOA algorithm for solving non-convex DED problem considering valve-point loading effects, ramp-rate limits, prohibited operating regions and spinning reserve capacity. The new method is shown to be statistically significantly better on a test systems consisting of ten generating units. The results obtained through the proposed method are compared with those reported in the literature.

III. DYNAMIC ECONOMIC DISPATCH PROBLEM FORMULATION

The classic DED problem minimizes the following incremental cost function associated to dispatchable units

$$\text{Min } F = \sum_{t=1}^T \sum_{i=1}^N F_{it}(P_{it}) \quad (\$) \quad (1)$$

where F is the total generating cost over the whole dispatch

period, T is the number of intervals in the scheduled horizon, N is the number of generating units, and $F_{it}(P_{it})$ is the fuel cost in terms of its real power output P_{it} at time t. Taking into account of the valve-point effects, the fuel cost function of i^{th} thermal generating unit is expressed as the sum of a quadratic and a sinusoidal function in the following form

$$F_{it}(P_{it}) = a_i P_{it}^2 + b_i P_{it} + c_i + |e_i \sin(f_i(P_{i\min} - P_{it}))| \quad (2)$$

$$J = F_{it}(P_{it}) \quad (3)$$

$$JF = 1/J \quad (4)$$

where a_i , b_i and c_i are cost coefficients, e_i and f_i are constants from the valve point effect of the i^{th} generating unit, and P_i is the power output of the i^{th} unit in megawatts. The minimization of the generation cost is subjected to the following equality and inequality constraints

1) Real power balance constraint

$$\sum_{i=1}^N P_{it} - P_{Dt} - P_{Lt} = 0 \quad (5)$$

where $t = 1, 2, \dots, T$. P_{Dt} is the total power demand at time t in megawatts and P_{Lt} is the transmission power loss at time t in megawatts. P_{Lt} is calculated using the B-Matrix loss coefficients and the general form of the loss formula using B-coefficients is

$$P_{Lt} = \sum_{i=1}^N B_{ii} P_{it}^2 \quad (6)$$

2) Real power generation limit

$$P_{i\min} \leq P_{it} \leq P_{i\max} \quad (7)$$

where $P_{i\min}$ is the minimum limit, and $P_{i\max}$ is the maximum limit of real power of the i^{th} unit in megawatts.

3) Generating unit ramp rate limits

$$P_{it} - P_{i(t-1)} \leq UR_i, \quad i = 1, 2, 3, \dots, N \quad (8)$$

$$P_{i(t-1)} - P_{it} \leq DR_i, \quad i = 1, 2, 3, \dots, N$$

where UR_i and DR_i are the ramp-up and ramp-down limits of i^{th} unit in megawatts. Thus the constraint of (8) due to the ramp rate constraints is modified as

$$\max(P_{i\min}, P_{i(t-1)} - DR_i) \leq P_{it} \leq \min(P_{i\max}, P_{i(t-1)} + UR_i) \quad (9)$$

such that

$$P_{it,\min} = \max(P_{i\min}, P_{i(t-1)} - DR_i) \quad \text{and} \quad (10)$$

$$P_{it,\max} = \min(P_{i\max}, P_{i(t-1)} + UR_i)$$

4) Prohibited operating zones

$$P_{it,\min} \leq P_i \leq P_{i,1}^L \quad (11)$$

$$P_{i,q-1}^U \leq P_i \leq P_{i,q}^L$$

$$P_{i,ni}^U \leq P_i \leq P_{i,\max}, q = 2, 3, \dots, n_i$$

where

i = unit index

n_i = the number of prohibited zones for unit i .

q = the index of prohibited zones for unit i .

$P_{i,q}^L$ = the lower boundary of q^{th} prohibited zone for unit i .

$P_{i,q}^U$ = the upper boundary of q^{th} prohibited zone for unit i .

5) System spinning reserve constraint

$$\sum_{i=1}^N S_{it} \geq S_{Rt} \quad (12)$$

$$S_{it} = \min\{(P_{it\max} - P_{it}), S_{i\max}\}$$

where S_{it} = reserve contribution of the i^{th} unit at t^{th} interval

S_{Rt} = spinning reserve demand at t^{th} interval in megawatts

$S_{i\max}$ = maximum reserve contribution of i^{th} unit

IV. AN OVERVIEW OF INTEGRATED TECHNIQUES

The Particle Swarm Optimization was introduced by James Kennedy and Russell C. Eberhart in 1995 [26]. PSO is motivated from the simulation of the behavior of social systems such as fish schooling and birds flocking. It is similar to other evolutionary computation techniques like Genetic Algorithm (GA) and Evolutionary Programming (EP) in that PSO also initializes a population of individuals randomly. These individuals are known as particles and have positions and velocities. Each particle adjusts its velocity dynamically corresponding to its flying experiences and its neighbors. The three main components that affect the changing of the velocity are internal, cognitive and social components. Internal component represents the particles behavior for moving in the previous direction. The cognitive component represents the memory of the particle for attracting to its best position ($pbest$) based on its own thinking. The social component represents the memory of the particle for attracting its previous best position among the group ($gbest$), here the particle changes its velocity based on the social-psychological adaptation of knowledge. So the velocity of the particle is updated to new position according to its modified velocity. The best evaluation function (fitness) is evaluated for the entire run and is stored in $gbest$. The velocity of i^{th} particle is represented as $V_i = (V_{i1}, V_{i2}, \dots, V_{ij})$. Hence the modified velocity of each particle can be calculated using the information.

- The current velocity
- The distance between the current position and $pbest$.
- The distance between the current position and $gbest$.

$$V_{ij}(it+1) = w * V_{ij}(it) + C_1 * rand_1 * (pbest_{ij} - x_{ij}(it)) + C_2 * rand_2 * (gbest_i - x_{ij}(it)) \quad (13)$$

where

$x_{ij}(it)$ = current position of the i^{th} particle in j^{th} dimension at iteration it .

$V_{ij}(it)$ = velocity of the i^{th} particle in j^{th} dimension at iteration it .

The i^{th} particle position can be represented as $x_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{iN})$ in the N-dimensional space.

The best previous position of the i^{th} particle is stored and represented as $x_{besti} = (x_{best,i1}, x_{best,i2}, \dots, x_{best,ij})$.

The updated position of each particle is expressed as

$$x_{ij}(it+1) = x_{ij}(it) + V_{ij}(it+1) \quad (14)$$

The new positions are calculated repeatedly till a pre-specified maximum number of iterations are reached.

Differential Evolution is one of the excellent evolutionary algorithms for global optimization developed by Price and Storn in 1995 [27]. It is a simple population based, stochastic parallel search evolutionary algorithm developed for nonlinear and non-differentiable continuous space functions in order to attain global optimization. DE is a robust statistical method for cost function minimization, which does not make use of a single parameter vector but instead uses a population of equally important vectors. To generate a new generation the evolutionary operator's mutation, recombination, and selection are applied to every population member. In each generation to change each population member X_i , a donor vector v_i is created. Three parameter vectors $r1$, $r2$, and $r3$ are chosen in a random fashion from the current population. A scalar number sf scales the difference of any two of the three vectors and the scaled difference is added to the third one. Donor vector for j^{th} component of k^{th} population at G generation is expressed as

$$v_{i,j}(G+1) = X_{r1,j}(G) + sf * (X_{r2,j}(G) - X_{r3,j}(G)) \quad (15)$$

where $j = 1, 2, \dots, D$

The mutation factor sf is a control parameter which scales the difference of any two of the three vectors and the scaled difference is added to the third one. This step is known as differential variation and the result is known as the mutant vector. DE is sensitive to the choice of mutation factor and to avoid search stagnation it is usually taken in the range 0.1 to 1. In recombination a trial vector is generated by replacing certain parameters of the target with the corresponding parameters of a randomly selected donor vector. The crossover constant CR is taken in the range [0, 1].

$$u_{i,j}(G+1) = v_{i,j}(G+1) \text{ if } rand(0,1) < CR$$

$$u_{i,j}(G+1) = X_{i,j}(G) \text{ otherwise} \quad (16)$$

The evaluated fitness value of population vector and the

trial vector are compared and the best is selected as in equation (16). Hence the fitness of each trial vector $u_{ij}(G+1)$ is compared with that of its parent target vector $X_{ij}(G)$ and the maximum fitness or minimum cost one is allowed to proceed to the next generation.

Bacterial foraging algorithm was proposed by Passino in 2002 [28], the BFO approach was based upon the fact of survival of bacteria especially *Escherichia coli*, in the natural changing environment. The process depends upon their fitness criteria which depend upon their food searching and motile behavior. The law of evolution supports the bacteria with good foraging strategies and eliminates or reshapes those with poor strategies. The genes of bacteria with good foraging strategies are propagated in evaluation chain since they possess ability to reproduce even better bacteria in future generations. The foraging strategy of *Escherichia coli* bacteria is explained by three steps chemo-tactic, reproduction and elimination-dispersal with the help of which global searching capability of this algorithm is executed. Chemo-tactic process of bacterial foraging describes the movement of an *E.coli* cell through swimming and tumbling via flagella. It can move in two different ways, it can swim for a definite period of time or it can tumble and it alternates between these two modes of operation for the entire lifetime. For example if θ_k denotes the k^{th} bacterium and CS_k is the size of the step taken in the random direction specified by the tumble.

$$\begin{aligned}\theta_k &= \theta_k + CS_k * \Delta_k \\ \Delta_k &= \frac{\Delta_k}{\sqrt{\Delta_k^T \cdot \Delta_k}}\end{aligned}\quad (17)$$

where Δ is a unit length vector in the random direction

In this article we come up with a hybrid optimization technique, which synergistically couples the BFOA with the PSO and DE. The proposed algorithm performs local search through the chemo-tactic movement operation of BFOA whereas the global search over the entire search space is accomplished by a PSO-DE operator. In this way it balances between exploration and exploitation enjoying best of both the worlds.

V. BACTERIAL FORAGING PSO-DE ALGORITHM FOR DED PROBLEM

The steps for implementation of the Bacterial Foraging and PSO-DE optimization are given below.

Step1: Initializing parameters

1a) Randomly initialize positions $P(t, j, k, l, m)$ and velocities $V(t, j, k)$ in t^{th} hour for j^{th} unit, k^{th} bacterial population, l^{th} reproduction ($l = 1$) and m^{th} elimination-dispersal ($m = 1$).

1b) Define each particle best as $pbest$ and the best position

of all particles as $gbest$.

Step2: Starting of the elimination-dispersal loop ($m = m+1$).

Step3: Starting of the reproduction loop ($l = l+1$).

Step4: Starting of the chemo-tactic loop ($iter = iter+1$).

Step5: For each bacteria ($k = k+1$).

Step6: Compute fitness function $JF(k, l, m)$ and cost function $J(k, l, m)$, let

$$JF_{last} = JF(k, l, m) \quad (18)$$

$$JF = \frac{1}{J + \text{penalty function}} \quad (19)$$

$$\text{penalty function}(k) = \begin{cases} k_c |(P_k - P_{k, \lim})| & \text{if violated} \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

where k_c is a constant value and $P_{k, \lim}$ is the specified boundary limits

Step7: Move to new position

$$7a) \text{ Compute } CS(k) = rand * k_d \quad (21)$$

$$7b) P(t, j, k, l, m) = P(t, j, k, l, m) + CS(k) * \Delta(t, j, k) \quad (22)$$

where $\Delta(t, j, k) = V(t, j, k)$ for $k=1, 2, 3, \dots, N_b$

7c) Handle constraints

- generation limit
- ramp rate limit
- prohibited operating zones
- system spinning reserve

7d) Compute fit and cost- JF and J

Step8: swimming

8a) $swim - count = 0$

8b) while ($swim - count < N_s$) and $JF(k, l, m) > JF_{last}$

I. $swim - count = swim - count + 1$

II. $JF_{last} = JF(k, l, m)$

III. $P(t, j, k, l, m) = P(t, j, k, l, m) + CS(k) * \Delta(t, j, k)$

IV. Handle constraints

- generation limit
- ramp rate limit
- prohibited operating zones
- system spinning reserve

V. Compute JF and J

Step9: Differential Evolution

9a) while ($rand < CR$)

$$\begin{aligned}P_{def}(t, j, k, l, m) &= P(t, j, k, l, m) + \\ &sf * (P(t, j, r_2, l, m) - P(t, j, r_3, l, m))\end{aligned} \quad (23)$$

9b) Handle constraints

- generation limit
- ramp rate limit
- prohibited operating zones
- system spinning reserve

9c) Compute cost and fit- J_{def}, JF_{def}

Step10:

if ($JF_{def} > JF$) *then*

$JF = JF_{def}, P = P_{def}, J = J_{def}$

Step11: Update best location $pbest$ and $gbest$ for JF, J and P .

Step12: 12a) Compute new velocity

$$\begin{aligned} V(t, j, k) &= w * V(t, j, k) \\ &+ C_1 * rand * (pbest(t, j, k) - P(t, j, k, l, m)) \\ &+ C_2 * rand * (gbest(t, j) - P(t, j, k, l, m)) \end{aligned} \quad (24)$$

12b) $\Delta = V$

Step13: If $k < S$ then go to step5

Step14: If $iter < N_c$ then go to step4

Step15: Reproduction operations

15a) JF_{health} is the measure of particles with good fitness (good fitness means higher fitness and lower cost)

$$JF_{health}(k) = pbestJF(k, l, m) \quad (25)$$

where $pbestJF$ is the population of best fitness of the current generation

15b) Sort bacteria in the order of ascending fit. The weaker $N_{br} = N_b / 2$ bacteria die and the rest N_{br} best bacteria each split into two bacteria which are placed at the same location.

15b) If $l < N_{re}$ then go to step3 to start the next generation in the chemo-tactic loop else go to step16.

Step16: Elimination-dispersal operation

16a) For $k=1, 2, 3, \dots, N_b$ a random number is generated and if it is less than or equal to P_{ed} than that bacterium is dispersed to a new random location else it remains at its original location.

16b) If $m < N_{ed}$ then go to step2 otherwise stop.

In the original bacterial foraging algorithm CS_k indicate a basic chemo-tactic step size (chemo-tactic step size is the lengths of steps taken during runs). This value should be greater than zero. It is taken in the tumble and swimming operations so that it moves in a direction specified by them and helps in reduction of cost. The normal choice for this step size is 0.1. The chemo-tactic step size in this paper is proposed to vary in the range (approximately 0.1 to 4.5) by using the relation

$$CS(k) = rand(0,1) * k_d \quad (26)$$

Usually the k_d value is a number between 1 and 5; $rand(0, 1)$ is a uniformly distributed random number within the range $[0, 1]$. For every chemo-tactic step in the tumble and swimming operation the step size value taken in this paper lead to fine reduction in cost.

VI. RESULTS AND DISCUSSIONS

The feasibility of the method BPSO-DE is demonstrated on a test system (ten-unit) for scheduled time duration one day divided into 24 intervals. The spinning reserve is required to be greater than or equal to 5% of the load demand at every dispatch interval. The proposed method satisfies the constraint load demand in each interval, prohibited operating zones, system spinning reserve and also dynamic constraint ramp-rate constraint. An experimentation is done for different values of control parameters such as C_1, C_2, CR, sf and k_d . It is observed that for $C_1=0.1, C_2=0.1, CR=0.8$ and $sf=0.1$ and $k_d=5$ the convergence leads to better cost.

Example: ten-unit system

Ten-unit system data such as cost coefficients, generation limits, ramp-rate limits, spinning reserve maximum limit for each unit and load demand in each interval is taken from [25]. The parameters chosen for this system are $N_c = 500, N_{ed}=2, N_{re}=2, N_s=4, N_b=120$ and $P_{ed}=0.2$. The convergence characteristic are plotted for number of evaluation (number of evaluation = $N_c * N_{ed} * N_{re}$) and total generation cost is as in Figure 1, Figure 2 represents the cost for each hour and Figure 3 compares the spinning reserve per hour with 5% load demand.

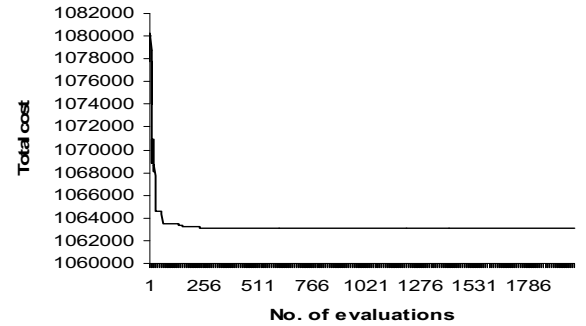


Fig. 1. Cost convergence characteristics for ten-unit system

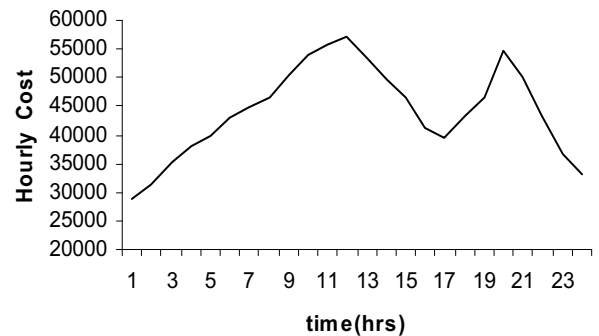


Fig. 2. Hourly cost curve for ten-unit system

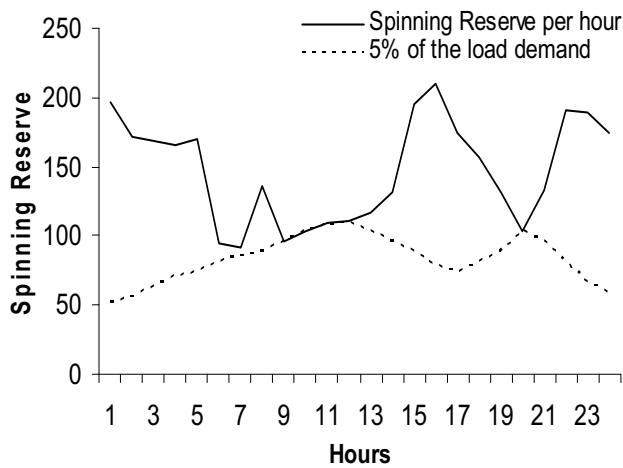


Fig. 3. Comparison of spinning reserve with 5% of the load demand for each hour

From simulation it is observed that the proposed method reduces the total generation cost when compared with other heuristic methods like HDE and VSHDE as shown on Table I. The optimum scheduling of generating units for 24 hours using the proposed method is tabulated in Table II. Computation time of the proposed method per a chemo-tactic step is 7.065seconds.

TABLE I
COMPARISON OF RESULTS FOR TEN-UNIT SYSTEM

Method	HDE [25]	VSHDE [25]	BPSO-DE
Fuel Cost (\$)	1343243.2	1100666.4	1063048

VII. CONCLUSION

In this article a new heuristic technique is developed by integrating BFOA, PSO and DE techniques for solving DED problem. PSO-DE operators work effectively when combined with BFOA and leads to reach global solutions. The efficiency of the proposed BPSO-DE method for solving non-convex DED problem considering prohibited operating zones, valve-point loading effect and spinning reserve is observed from its solutions and compared with the results obtained from heuristic methods like HDE and VSHDE. The comparative study reveals the superiority and feasibility of BPSO-DE technique to solve the DED problem. Due to its superiority in reaching good quality solution this method in future work can be extended to solve for security constraints.

VIII. REFERENCES

- [1] G.P.Granelli, P.Marannino, M.Montagna, and A.Silvestri, "Fast and efficient gradient projection algorithm for dynamic generation dispatching," *Proc. Inst. Elect. Eng., Gener. Transm. Distrib.*, vol. 136, no. 5, pp.295-302, Sep. 1989.
- [2] F.Li, R.Morgan, and D.Williams, "Hybrid genetic approaches to ramping rate constrained dynamic economic dispatch," *Elect. Power Syst. Res.*, vol. 43, no. 2, pp. 97-103, Nov. 1997.
- [3] X.S.Han, H.B.Gooi, and Daniel S.Kirschen, "Dynamic economic dispatch: feasible and optimal solutions," *IEEE Trans. Power Syst.*, vol. 16, no.1, pp. 22-28, Feb. 2001.
- [4] T.Aruldas Albert Victoire, and A.Ebenezer Jeyakumar, "Deterministically guided PSO for dynamic dispatch considering valve-point effect," *Elect. Power Syst. Res.*, vol. 73, no.3, pp.313-322, 2005
- [5] D.C.Walters and G.B.Sheble, "Genetic algorithm solution of economic dispatch with valve point loadings," *IEEE Trans. Power Syst.*, vol. 8, no.3, pp.1325-1331, Aug. 1993.
- [6] T.Jayabharathi, K.Jayaprakash,N.Jeyakumar, and T.Raghunathan,"Evolutionary programming techniques for different kinds of economic dispatch problems," *Elect. Power Syst. Res.*, vol. 73, no. 2, pp.169-176, Feb. 2005.
- [7] Zwe-Lee Gaing, "Particle swarm optimization to solving the economic dispatch considering the generator constraints," *IEEE Trans. Power Syst.*, vol. 18, no.3, pp. 1187-1195, Aug. 2003.
- [8] B.K.Panigrahi, V.Ravikumar Pandi, Sanjoy Das, "Adaptive Particle Swarm Optimization approach for static and dynamic economic load dispatch", *Energy Conversion and Management* 49(2008), pp. 1407-1415.
- [9] Pichet Sriyanyong, "A Hybrid Particle Swarm Optimization Solution to Ramping Rate Constrained Dynamic Economic Dispatch", *World Academy of Science, Engineering and Technology* 47, 2008.
- [10] Nima Amjadi and Hadi Nasiri-Rad, "Non convex economic dispatch with AC constraints by a new real coded genetic algorithm", *IEEE Trans. Power Syst.*, vol. 24 no. 3, pp. 1489-1502, Aug. 2009.
- [11] Jong-Bae Park, Yun-Won Jeong, Joong-Rin Shin and Kwang Y. Lee, "An improved Particle Swarm Optimization for non convex economic dispatch problems", *IEEE Trans. Power Syst.*, vol. 25, no. 1, pp. 156-166, Feb. 2010.
- [12] K.S.Hindi and M.R.Ab Ghani, "Dynamic economic dispatch for large scale power systems: a Lagrangian relaxation approach," *Elect. Power Syst. Res.*, vol. 13, no. 1, pp. 51-56, 1991.
- [13] D.C.Walters and G.B.Sheble, "Genetic algorithm solution of economic dispatch with valve-point loadings," *IEEE Trans. Power Syst.*, vol. 8, no.3, pp.1325-1331, Aug. 1993.
- [14] H.T.Yang, P.C.Yang, and C.L.Huang, "Evolutionary programming based economic dispatch for units with non-smooth incremental fuel cost function," *IEEE Trans. Power Syst.*, vol. 11, no.1, pp.112-118, Feb. 1996.
- [15] N.Sinha, R.Chakrabarti, and P.K.Chattopadhyay, "Evolutionary programming techniques for economic load dispatch," *IEEE Trans. Evol. Comput.*, vol. 7, no.1, pp. 83-94, Feb. 2003.
- [16] K. Shanti Swarup, Anand Natarajan, "Constrained Optimization using Evolutionary Programming for Dynamic Economic Dispatch", *Proceedings of ICISIP 2005*.
- [17] K.P.Wong and Y.W.Wong, "Genetic and genetic/simulated-annealing approaches to economic dispatch," *Proc. Inst. Elect. Eng., Gener. Transm. Distrib.*, vol. 141, no. 5, pp.507-513, Sept. 1994.
- [18] C.K.Panigrahi, P.K.Chattopadhyay, R.N.Chakrabarti, and M.Basu, "Simulated annealing technique for dynamic economic dispatch," *Electric power components and systems*, vol. 34, no.5, pp.577-586, May 2006.
- [19] R.Balamurugan, S.Subramanian, "An Improved Differential Evolution Based Dynamic Economic Dispatch with Nonsmooth Fuel Cost Function", *Journal of Electrical Systems* 3-3(2007):151-16
- [20] Hao, Z. F, Guo, G. H., Huang, H., "A Particle Swarm Optimization Algorithm with Differential Evolution", in *IEEE Int.Conf. Systems, man and Cybernetics*, Aug. 2007, Vol. 2, pp. 1031-1035.
- [21] D.Attaviriyanupap, H.Kita, E.Tanaka, and J.Hasegawa, "A hybrid EP and SQP for dynamic economic dispatch with nonsmooth incremental fuel cost function," *IEEE Trans. Power Syst.*, vol. 17, no.2, pp.411-416, May 2002. .
- [22] T.Aruldas Albert Victoire, and A.Ebenezer Jeyakumar, "Reserve constrained dynamic dispatch of units with valve-point effects," *IEEE Trans. Power Syst.*, vol. 20, no.3, pp.1273-1282, Aug. 2005.
- [23] S. Titus and A. Ebenezer Jeyakumar, "A Hybrid EP-PSO-SQP Algorithm for Dynamic Dispatch Considering Prohibited Operating Zones", *Electrical Power Components and Systems*, vol. 36, no. 5, pp. 449-467, May 2008.
- [24] W. G. Wood, "Spinning reserve constrained static and dynamic economic dispatch", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-101, No. 2 February 1982.
- [25] J.P.Choiu, "A variable scaling hybrid differential evolution for solving large-scale power dispatch problems", *IET Generation, Transmission & Distribution*, Vol. 3, Iss. 2, pp. 154-163, 2009

- [26] Kennedy,J, Eberhart,R, "Particle swarm optimization", *In Proceedings of IEEE International Conference on Neural Networks*, (1995) 1942-1948.
- [27] R.Storn,and K.Price, "Differential evolution – A simple and efficient heuristic for global optimization over continuous space," *J. Global Optimization*, 11, pp.341-359, 1997.
- [28] Passino.K.M, "Biomimicry of Bacterial Foraging for Distributed Optimization and Control", *IEEE Control Systems Magazine*, 52-67, (2002).
- [29] Praveena, P, Vaisakh, K, S R M Rao, "A Bacterial Foraging and PSO-DE Algorithm for Solving Dynamic Economic Dispatch Problem with Valve-Point Effects," *Integrated Intelligent Computing (ICIIC)*, 2010 First International Conference on , vol., no., pp.227-232, 5-7 Aug. 2010.
- [30] G. Acampora, M. Gaeta, and V. Loia, "Exploring e-learning knowledge through ontological memetic agents," *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 66-77, 2010.
- [31] Ling-Ning Xing, Ying-Wu Chen, Ke-Wei Yang, Feng Hou, Xue-Shi Shen, Huai-Ping Cai, "A hybrid approach combining an improved genetic algorithm and optimization strategies for the asymmetric traveling salesman problem", *Engineering Applications of Artificial Intelligence*, Volume 21, Issue 8, Pages 1370-1380, December 2008.
- [32] G. Acampora, M. Gaeta, and V. Loia, "Hierarchical optimization of personalized experiences for e-learning systems through evolutionary models," *Neural Comput. Applicat.*, 2009.
- [33] Huizhi Yang, Jianguo Shi, "A Hybrid Approach Based on Immune Genetic Algorithm and Integer Linear Programming for the Container Loading Problem," *Education Technology and Computer Science*, International Workshop on, pp. 323-326, 2010 Second International Workshop on Education Technology and Computer Science, 2010.
- [34] Haiyan Zhao, Zhili Pei, Jingqing Jiang, Renchu Guan, Chaoyong Wang and Xiaohu Shi, "A Hybrid Swarm Intelligent Method Based on Genetic Algorithm and Artificial Bee Colony ," *Lecture Notes in Computer Science*, 2010, Volume 6145/2010, pp. 558-565, DOI: 10.1007/978-3-642-13495-1_68.
- [35] Xiaohui Yuan, Liang Wang, Yongchuan Zhang, Yanbin Yuan, "A hybrid differential evolution method for dynamic economic dispatch with valve-point effects", *Expert Systems with Applications*, Volume 36, Issue 2, Part 2, Pages 4042-4048, March 2009.
- [36] T.S. Chung, T.W. Lau, "Hybrid PSO and DE approach for dynamic economic dispatch with non-smooth cost functions", *International Journal of Modelling, Identification and Control* 2009 - Vol. 8, No.4 pp. 317 - 326.

TABLE II
BEST SCHEDULING OF TEN-UNIT SYSTEM (BPSO-DE)

Unit Hour	1 (MW)	2 (MW)	3 (MW)	4 (MW)	5 (MW)	6 (MW)	7 (MW)	8 (MW)	9 (MW)	10 (MW)	Loss (MW)
1	303.3107	135.0954	76.1023	120.4930	73.0667	121.9583	93.1576	47.0012	20.0088	55.0000	9.1940
2	303.1047	135.0172	144.0161	118.1940	122.2406	112.6885	63.4516	47.0031	20.0454	55.0000	10.7612
3	303.2802	184.8563	185.0678	159.1366	122.8930	121.9701	43.0133	47.0035	49.8917	55.0000	14.1126
4	326.5068	225.1377	256.8599	181.0907	172.7197	121.9962	20.0175	47.0058	20.0190	55.0000	20.3533
5	303.1970	305.0000	282.5101	180.7022	138.0052	121.6518	20.0278	47.0021	49.9778	55.0000	23.0741
6	379.8711	305.0000	315.5119	223.7387	173.8315	114.4116	20.0192	47.0063	22.7838	55.0000	29.1740
7	456.7183	305.0000	339.8606	244.7551	124.4338	121.9990	20.0141	47.0026	20.0104	55.0000	32.7938
8	453.9357	385.0000	273.5526	231.5744	172.2982	121.6684	20.0609	47.0016	49.3123	55.0000	33.4041
9	420.8934	400.2324	322.5569	241.3323	221.1822	122.0000	49.9804	76.9809	52.0260	55.0000	38.1845
10	428.5658	459.9303	339.9960	242.4761	223.2954	159.9182	69.4463	85.2107	52.0261	55.0000	43.8649
11	456.7193	459.9644	307.7184	257.7640	241.5928	159.9996	93.0807	85.3439	73.5565	55.0000	44.7396
12	423.2593	450.9271	340.0000	295.7597	242.9963	159.9999	93.1312	85.2921	79.6214	55.0000	46.9870
13	443.5317	384.5473	298.1866	299.9900	242.9553	159.4991	93.0246	85.2413	51.6094	55.0000	41.5853
14	428.0703	304.9407	297.3060	299.8588	222.5140	122.0000	93.0787	85.2300	52.0471	55.0000	36.0456
15	379.8416	304.6283	297.2481	250.0262	222.9617	121.9576	82.0084	71.5163	22.5337	55.0000	31.7219
16	379.9198	225.2108	257.4714	241.0372	174.2891	121.9534	56.5162	47.0023	20.0123	55.0000	24.4125
17	379.6118	303.6819	197.7146	198.3923	126.2865	121.8149	52.2891	47.0037	20.0073	55.0000	21.8022
18	379.7567	381.9008	261.5448	237.4783	131.8385	120.6646	22.2964	47.0024	20.0075	55.0000	29.4899
19	380.2061	396.7896	286.3196	234.7688	176.9342	121.9344	50.5152	57.3339	49.0884	55.0000	32.8901
20	420.7247	459.9857	339.9737	248.6047	223.7195	159.1565	55.8475	77.7104	75.0658	55.0000	43.7885
21	379.8003	380.0543	285.7954	297.2375	222.3685	156.7065	84.6191	47.7181	52.0418	55.0000	37.3417
22	379.0003	304.2104	231.2800	247.4262	192.0941	121.9266	54.7050	47.0063	22.1893	55.0000	26.8381
23	316.3778	225.3004	188.8902	203.6083	146.5196	121.9951	24.7158	47.0330	20.0304	55.0000	17.4708
24	303.1660	146.7164	172.0509	157.8998	122.8888	121.9967	20.0078	47.0042	49.9400	55.0000	12.6706