Solution of Dynamic Economic Emission Dispatch Problem by Hybrid Bacterial Foraging Algorithm


Abstract—In this paper, a new heuristic Bacterial Foraging PSO-DE (BPSO-DE) algorithm is proposed for solving dynamic economic emission dispatch (DEED) problem. The DEED problem is a multi objective optimization problem with nonlinear constraints. The algorithm such as BFOA, Particle Swarm Optimization (PSO) and Differential Evolution (DE) are synergistically coupled to form the new hybrid optimization algorithm BPSO-DE. The PSO operator is used in the process of updating BFOA to generate bacteria with good foraging strategies and DE operator fine tunes the solution achieved through bacterial foraging and PSO algorithm. To demonstrate the efficacy of the proposed algorithm a 10-unit test system is considered for illustrative purpose.

Keywords— Bacterial foraging optimization Algorithm, Dynamic economic emission dispatch, differential evolution, particle swarm optimization

I. INTRODUCTION

The fundamental objective of dynamic economic dispatch (DED) problem of electric power generation is to schedule the committed generating unit outputs in order to meet the predicted load demand with minimum operating cost, while satisfying all system inequality and equality constraints [1], [2]. Therefore, the DED problem is a highly constrained large-scale nonlinear optimization problem. The valve-point effect introduces ripples in the heat-rate curves and make the objective function non-convex, discontinuous, and with multiple minima [3], [4], [5]. The fuel cost function with valve point loadings in the generating units is the accurate model of the DED problem. The DEED problem is an extension to traditional static economic power dispatch problem in which the mechanical constraint is introduced to avoid shortening the lifetime of turbines and boilers. In order to keep the thermal gradients of equipment within the safe limits, the mechanical constraint is transformed into a limit on rate of increase in electrical output of every dispatchable unit [6], [7], [8]. Hence the unit manufacturer should design the ramp rate limits, so that the output of a unit will not ramp up or ramp down in every time interval more than a few megawatts, as specified by the unit’s manufacturer.

Nowadays strategically utilizing available resources and achieving electricity at cheap rates without sacrificing the social benefits is of major significance [9], [10]. The environmental pollution plays a major role as it had a major threat on the human society, hence it became compulsory to deliver electricity at a minimum cost as well as to maintain minimum level of emissions. Lowest emissions are considered as one of the objectives in combined economic and emission dispatch problems, along with cost economy. Atmospheric pollution due to release of gases such as nitrogen oxides (NOx), carbon dioxide (CO2), and sulphur oxides (SOx) into atmosphere by fossil-fuel based electric power stations affects not only humans but also other forms of life such as birds, animals, plants and fish, while causes global warming too [11], [12].

The dispatching of emission is a short-term option where the emission, in addition to fuel cost objective, is to be optimized. Thus, DEED problem can be handled as a multi-objective optimization problem and requires only small modification to include emission. Hence, the DEED problem can be converted to a single objective problem by linear combination of various objectives using different weights. The important characteristic of the weighted sum method is that different pareto-optimal solutions can be obtained by varying the weights [13].

The BFOA is gaining recognition in the community of scientists, for its efficiency in solving certain complex real-world optimization problems. BFOA is determined based on the foraging strategies of Ecoli bacterium cells[13,14]. The PSO and DE algorithms are excellent heuristics like any other evolutionary algorithms [13, 15]. In the literature it is observed that these algorithms reach stagnation after certain number of iterations and they will stop to proceed towards global optimal solutions. Hence in this paper a hybrid approach BPSO-DE involving BFOA, PSO and DE algorithms is proposed for solving DEED problem considering valve-point effects and ramp-rate limits [4,13]. The new technique showed statistically better on a 10 unit test system. The results obtained by the proposed BPSO-DE approach are compared with those methods reported in the literature [13,16, 17].
II. NOMENCLATURE

\( N_b \) Number of the bacterial population [16,17].

\( N_C \) Number of the chemo-tactic steps.

\( N_{re} \) Number of the reproduction steps.

\( N_{ed} \) Number of the elimination-dispersal events.

\( N_S \) Swimming length which represent the maximum number of steps taken by each bacterium during movement from low nutrient area to high nutrient area.

\( N \) Number of real power generating units

\( P_{ed} \) Probability of the elimination and dispersal loop

\( CS \) Step size taken in random direction specified by the tumble.

\( P(t, j, k, l, m) \) Position vector of \( j^{th} \) unit in \( t^{th} \) hour, \( k^{th} \) bacterial population, \( l^{th} \) reproduction and \( m^{th} \) elimination-dispersal, where \( k = 1,2,...,S \), \( l = 1,2,...,N_{re} \) and \( m = 1,2,...,N_{ed} \)

\( Vel(t, j, k) \) Velocity vector of the \( j^{th} \) unit in \( t^{th} \) hour and \( k^{th} \) Bacterial population.

\( p_{best} \) Best previous position of \( k^{th} \) particles.

\( g_{best} \) The index of the gbest among all particles in the population

\( sf \) Scale factor

\( CR \) Cross over probability

\( C_1, C_2 \) Acceleration constants

\( Rand \) Random value in the range \([0,1]\)

\( JF(k,l,m) \) Fitness value of \( k^{th} \) bacterial population, \( l^{th} \) reproduction and \( m^{th} \) elimination-dispersal[16,17]

\( r_2, r_3 \) Parameter vectors chosen randomly from the current population.

III. PROBLEM FORMULATION

The DEED problem is a multi-objective problem which minimizes both emission and cost simultaneously, along with the inequality and equality constraints. The objectives considered in the formulation of the DEED problem are:

(a) Cost: The function for cost of generation considering the valve-point effect can be expressed as the sum of a quadratic and a sinusoidal function. The total fuel cost function in terms of real power output of generators is expressed as

\[
f_1 = \sum_{i=1}^{N} \sum_{t=1}^{T} \left[ a_i P_{a,i}^2 + b_i P_{a,i} + c_i + d_i \sin\left( e_i (P_{a,i} - P_{a,0})\right) \right]
\]

where the constant \( a_i \), \( b_i \) and \( c_i \) represents generator cost coefficients and \( d_i \) and \( e_i \) represents the valve point effect coefficients of the \( i^{th} \) generator.

(b) Emission: The total emission function is represented by the combination of quadratic and exponential functions and is expressed as

\[
f_2 = \sum_{i=1}^{N} \sum_{t=1}^{T} \left[ \alpha_i P_{a,i}^2 + \beta_i P_{a,i} + \gamma_i + \eta_i \exp(\delta_i P_{a,i}) \right]
\]

where \( \alpha_i, \beta_i, \gamma_i, \eta_i, \delta_i \) represents the emission coefficients of the \( i^{th} \) generating unit.

The minimization of the above two objective function is subjected to the following inequality and equality constraints:

(i) Load power balance constraint

The total load plus loss in the transmission lines met by the real power generation at each time interval over the scheduling horizon is expressed as

\[
\sum_{i=1}^{N} P_{a,i} - P_{in} - P_{lt} = 0
\]

where \( t = 1, 2, ..., T \). \( P_{in} \) is the total real power load at time \( t \) and \( P_{lt} \) is the transmission real power loss at time \( t \).

The real power loss \( P_{lt} \) is calculated using the B-loss coefficients and is given by

\[
P_{lt} = \sum_{i=1}^{N} \sum_{j=1}^{N} P_{B,i} P_{j} P_{r}
\]

(ii) Real power operating limits:

\[
P_{i, min} \leq P_i \leq P_{i, max} \quad i \in N, t \in T
\]

where \( P_{max} \) and \( P_{min} \) are maximum and minimum limits of the real power output of the \( i^{th} \) generating unit.

(iii) Generating unit ramp rate limits

\[
P_{i, (t-1)} - P_i \leq DR_i \quad \text{and} \quad P_{i, (t-1)} - P_i \leq UR_i
\]

where \( i = 1,2,3,....,N \)

\( DR_i \) and \( UR_i \) are the ramp-down and ramp-up limits of \( i^{th} \) generating unit in megawatts. Then the ramp rate constraints is expressed as

\[
\max(P_{i, (t-1)} - P_{i}), \min(P_{i, (t-1)} + UR_i) \leq P_{i}
\]

such that

\[
P_{i, max} = \max(P_{i, (t-1)} - DR_i) \quad \text{and} \quad P_{i, min} = \min(P_{i, (t-1)} + UR_i)
\]

In the proposed approach if the newly generated real power outputs of the generating units violates their boundary conditions(i.e if the total generation of all units cannot handle the load demand during that interval), then this constraint is handled in such a way that the load demand of that hour is satisfied. Similarly if the generation is not within the ramp-rate limits then repair those limits as given in equation (8) and this equation can also be expressed as
\[
\text{if } P_a > P_{a,\text{max}} \text{ then} \\
P_a = P_{a,\text{max}} \\
\text{if } P_a < P_{a,\text{min}} \text{ then} \\
P_a = P_{a,\text{min}} \\
\]

(9)

IV. PRINCIPLE OF MULTI-OBJECTIVE OPTIMIZATION

Many objective functions are optimized simultaneously which are non-commensurable, competing and conflicting objectives. The multi-objective optimization problem with such contradictory objective functions generally gives Pareto optimal solutions, instead of one optimal solution objectives. The multi-objective optimization problem with a number of objectives and several inequality and equality constraints can be formulated as follows:

Minimize \( f_i(y) \) \( i = 1, \ldots, N_{\text{obj}} \) \( \quad \) (10)
Subject to \( g_j(y) = 0 \) \( \quad \) (11)

For the purpose of comparison, the problem has been transformed to a single optimization problem weighted sum of generation cost function and emission objective function as follows.

Minimize \( w f_1 + (1-w)\lambda f_2 \) \( \quad \) (12)

Where, \( \lambda \) is the scaling factor and \( w \) is a variable weighting factor. Ten to eleven non-dominated solutions are generated and the algorithm has been applied ten or eleven times by varying \( w \).

V. OVERVIEW OF PARTICLE SWARM OPTIMIZATION

PSO is an evolutionary algorithm which initializes a population of individuals randomly as other evolutionary computational methods like GA and EP. These individuals are considered as particles having position and velocity. It searches for the optimum solution by updating generations, and population evolution is developed from previous generations. Because of its simplicity the PSO algorithm can be achieved with less computation time and memory. Each particle of PSO is moved throughout the problem space by following the current optimal particles. Each individual regulates its flying based on its own and its neighbors flying experience. According to its own thinking the particle attracts to its best position (pbest) and the particle adjusts its velocity according to its previous best position among the group (gbest). Therefore the particle velocity is revised to new position based on its current and new velocities, the distance between the pbest and current position and the distance between the gbest and current position as given in (13)

\[ V_{\text{el}_{j}}(\text{iter} + 1) = w_i \text{ Ve}_{l_{j}}(\text{iter}) + C_1 \text{rand}_i \ast (\text{pbest}_j - x_{ij}(\text{iter})) + C_2 \text{rand}_i \ast (\text{gbest}_j - x_{ij}(\text{iter})) \]

(13)

where
\( w_i \) = inertia weight factor
\( x_{ij}(\text{iter}) \) = current position of \( i^{\text{th}} \) particle in \( j^{\text{th}} \) dimension at iteration \( \text{iter} \)
\( V_{\text{el}_{j}}(\text{iter}) \) = velocity of \( i^{\text{th}} \) particle in \( j^{\text{th}} \) dimension at iteration \( \text{iter} \)

Suitable selection of inertia weight presents a balance between local and global explorations. The \( i^{\text{th}} \) particle position is represented as \( x_i = (x_{i1}, x_{i2}, x_{i3}, \ldots \ldots \ldots \ldots x_{iN}) \) in the \( N \)-dimensional space and the best previous position is stored as \( x_{ \text{best}_i} = (x_{\text{best}_{i1}}, x_{\text{best}_{i2}}, \ldots \ldots \ldots \ldots x_{\text{best}_{iN}}) \). If the local solution (pbest) has minimum cost when compared to the cost of the current global solution (gbest), then the local solution is replaced with best global solution. The updated position of each particle is expressed as

\[ x_{ij}(\text{iter} + 1) = x_{ij}(\text{iter}) + V_{\text{el}_{j}}(\text{iter} + 1) \] (14)

The new positions are evaluated repeatedly until a pre-defined maximum number of iterations are reached.

VI. OVERVIEW OF DIFFERENTIAL EVOLUTION OPTIMIZATION

Differential Evolution (DE) is a real coded, population based evolutionary algorithm developed by Storn and Price in 1995 [18] for global optimization. Like other evolutionary computational techniques it is a robust and simple algorithm. It starts by randomly choosing the initial candidate solutions within the boundary to explore the search space. Then the algorithm locates the global optimum solution by iterated refinement of the population through reproduction and selection. Hence it is a D-dimensional search variable vector with a population of pop. Cross over or mutation is used as a differential operator to produce new offspring from parent chromosomes. In each generation, a donor vector \( v_i \) is created to modify each population member \( X_i \). Three parameter vectors \( r1, r2, \) and \( r3 \) are chosen at random from the present 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 \( j^{\text{th}} \) component of \( k^{\text{th}} \) population at \( \text{iter} \) generation is expressed as

\[ v_{kj}(\text{iter} + 1) = X_{r1j}(\text{iter}) + sf \ast (X_{r2j}(\text{iter}) - X_{r3j}(\text{iter})) \] (15)

The CR is the “Crossover” constant and it is a control parameter of Differential Evolution. The crossover is performed on each of the \( D \) variables whenever a randomly picked number between 0 and 1 is within the CR value. The trial vector \( u_{kj} \) is outlined as

\[ u_{kj}(\text{iter}) = v_{kj}(\text{iter}) \quad \text{if rand}(0,1) < \text{CR} \]
\[ u_{kj}(\text{iter}) = X_{kj}(\text{iter}) \quad \text{otherwise} \] (16)
In the part of selection, the fitness value is obtain for both parent vector and the trial vector and the best is selected (16). So the value cost function of each trial vector \( u(t) \) is compared with that of its parent target vector \( X(t) \) and the one with lower cost is allowed to advance to the next iteration.

VII. OVERVIEW OF BACTERIAL FORAGING OPTIMIZATION

The BFOA is based upon the fact of survival of Escherichia coli bacteria in naturally varying environment. The procedure depends upon their fitness criteria and on foraging and motile behavior. The law of evolution supports those bacteria with good food searching strategies and eliminates or reshapes those with poor strategies. The bacteria genes with better foraging strategies are approved in evolution chain as they can reproduce better bacteria for future generations.

The foraging strategy of Escherichia coli bacteria consists of three essential steps namely chemotactic, reproduction and elimination–dispersal. Through these steps, the global searching ability of the algorithm is explored. Chemo-tactic process of bacterial foraging explains the motion of an E. coli cell through swimming and tumbling via flagella. During the entire lifetime, E. coli cell alternates between these two modes of motion. It can either swim for a specific period of time or it can tumble. For example, if \( \theta(k) \) denotes the \( k \)th bacterium and \( CS(k) \) is the size of the step taken in the random direction specified by the tumble.

\[
\theta(k) = \theta(k) + CS(k) \cdot \Delta(k) \\
\Delta(k) = \frac{\Delta(k)}{\sqrt{\Delta^2(k)}}
\]

where \( \Delta \) is a unit length vector in the random direction.

After swimming and tumbling operation of the BFA in reproduction process, the bacteria with worst foraging strategies are replaced by the bacteria with better foraging strategies. The elimination and dispersal of BFOA may placate bacteria very near to the good nutrient concentration.

VIII. BACTERIAL FORAGING PSO-DE ALGORITHM FOR SOLVING DEED PROBLEM (BPSO-DE)

The proposed technique solves the DEED problem by incorporating PSO operator in the BFOA algorithm. So the delta of BFA is replaced by the velocity vector in the PSO algorithm. From simulation it is observed that the operator of PSO algorithm makes the population of bacteria to arrive at a positive nutrient concentration (good fitness) at the earliest than the delta of BFA. Initially the real power outputs of the generating units are randomly generated. \( pbest \) is the population of best fitness of the current generation, hence update \( pbest \) and update \( gbest \) of fitness function for the randomly generated real power outputs. In the next step of Bacterial foraging PSO-DE algorithm the bacteria moves through swimming and tumbling operations by incorporating PSO operator and then fitness ‘ \( JF \) ’ is computed for these operations by including a penalty function in the fitness function. The penalty function eliminates those bacterial populations which exceeds their specified boundary limits.

In this paper the penalties are considered for the generation operating limits and ramp rate limits. After swimming and tumbling operation new bacterial population is generated with DE operator and corresponding fitness \( JFdef \) is computed. Both the fitness functions \( JF \) and \( JFdef \) are compared for every bacterial population and the best value is replaced in \( JF \), which means that if \( JFdef \) is greater than \( JF \) for any bacterial population then \( JF \) value is replaced with \( JFdef \) value. The best value of fitness, real power generation for the given time interval, cost and emission are updated for all the bacterial population as \( pbest \) and the best cost, emission, real power generation and fitness among the bacterial population are updated as \( gbest \). Now the velocity vector of PSO is updated for next chemo-tactic step.

After completion of definite number of chemotactic steps, reproduction loop is initialized at weaker bacterial population (i.e., bacteria with less fitness), and tenth half of the bacteria population is replaced with healthy bacteria. In the elimination-dispersal loop, the bacteria are dispersed to new random location if the presumed probability \( P_{ad} \) is either greater than or equal to the generated random number. The implementation steps of the BPSO-DE optimization algorithm are as given below.

**Step1:** Initializing parameters

1a) Initialize randomly the positions \( P(t, j, k, l, m) \) and velocities \( Vel(t, j, k) \) 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) For each particle define \( pbest \) and the best position of all particles as \( gbest \).

**Step2:** Starting of the elimination-dispersal loop \((m=m+1)\), where \( m=1, 2, 3, \ldots, N_{ad} \).

**Step3:** Starting of the reproduction loop \((l=l+1)\), where \( l=1, 2, 3, \ldots, N_{re} \).

**Step4:** Starting of the chemo-tactic loop \((iter=iter+1)\), where \( iter=1, 2, 3, \ldots, N_{c} \).

**Step5:** For each bacteria \((k=k+1)\), where \( k=1, 2, 3, \ldots, N_{b} \).

**Step6:** Compute fitness function \( JF(k, l,m) \) and cost function \( JC(k, l,m) \) and emission function \( JE(k, l,m) \) let

\[
JC = f_1 \\
JE = f_2 \\
J(k, l, m) = w \cdot JC(k, l, m) + \lambda \cdot (1 - w) \cdot JE(k, l, m)
\]

where \( \lambda = 10 \)

\[
JFlast = JF(k, l, m)
\]

\[
JF = \frac{1}{1 + \text{penalty function}}
\]
penalty function$(k) = \begin{cases} k \| P_i - P_{i,lim} \| & \text{if violated} \\ 0 & \text{otherwise} \end{cases} \quad (24)

where $k$ is a constant and $P_{i,lim}$ is the specified boundary limits.

**Step 7:** Move to new position
7a) Compute $CS(k) = rand + k_d$, where $k_d$ is a constant whose value is taken between 0.1 and 0.9.
7b) $P(t,j,k,l,m) = P(t,j,k,l,m) + CS(k) \cdot Delta(t,j,k)$, where $Delta(t,j,k) = Vel(t,j,k)$ for $k=1,2,3,\ldots,N_b$ and $j=1,2,3,\ldots,N$
7c) Now Handle constraints such as generation limits, and ramp rate limits
7d) Compute fitness- $JF$

**Step 8:** swimming
8a) $swim - count = 0 \quad (27)$
8b) while ($swim - count < N_j$) and $JF(k,l,m) > JF_{last}$

swim - count = swim - count + 1

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

$P(t,j,k,l,m) = P(t,j,k,l,m) + CS(k) \cdot Delta(t,j,k)$

Handle constraints

**Step 9:** Differential Evolution
9a) while ($rand < CR$)

$P_{def}(t,j,k,l,m) = P(t,j,k,l,m) + sf \cdot (P(t,j,r_{2l},l,m) - P(t,j,r_{1l},l,m)) \quad (29)$

9b) Handle constraints

**Step 10:**
if ($JF_{def} > JF$) then

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

$JE = JF_{def}, JC = JC_{def}$

**Step 11:** Update best location $p_{best}$ and $g_{best}$ for $JF, J, JC, JE$ and $P$.

**Step 12:** If $k < N_b$ then go to step 5

Step13:
13a) Compute new velocity

$Vel(t,j,k) = w_i \cdot Vel(t,j,k) + C_1 \cdot rand \cdot (p_{best}(t,j,k) - P(t,j,k,l,m))$

$+ C_2 \cdot rand \cdot (g_{best}(t,j) - P(t,j,k,l,m))$ \quad (31)

13b) $Delta = Vel$

**Step 14:** If $iter < N_i$ then go to step 4

**Step 15:** Reproduction operations
15a) $JF_{health}$ is the measure of particles with good fitness.

$JF_{health} = p_{best} JF(k,l,m) \quad (32)$

15b) Then sort bacteria in the ascending order of fitness. The weaker $N_w = N_b / 2$ bacteria die and the rest $N_w$ best bacteria each split into two bacteria which are placed at the same location.

15c) If $l < N_w$ then go to step 3 to start the next iteration in the chemo-tactic loop else go to step 16.

**Step 16:** Elimination-dispersal operation
16a) For $k = 1, 2, 3, \ldots, N_b$ a random number is generated and if the random number is less than or equal to $P_{ad}$ then that bacterium is dispersed to a new random location else it remains at its original location.

16b) If $m < N_{ad}$ then go to step 2 otherwise stop.

The algorithm (Step 1 to Step 16) has been applied ten times by varying the weighing factor $w$ ten times to obtain the pareto-optimal solution.

In the original BFA, $CS(k)$ indicate the length of steps considered during the runs as basic chemo-tactic step size. This value of the step should always be greater than zero and most of the times it is considered as 0.1. In tumble and swimming operations, chemo-tactic step size is used so that the bacteria move in a specified direction trying to reduce the cost. The chemo-tactic step size in this paper is proposed to vary in the range of (0.1, 1.8) using the relation

$CS(k) = rand(0,1) + k_d \quad (33)$

Where, $k_d$ is a constant in the range 0.1 to 0.9. For each chemo-tactic step in tumble and swimming operation, the step size value considered in this paper direct to fine reduction in cost.

**IX. SIMULATION RESULTS**

The feasibility of the proposed methods BPSO-DE is demonstrated on a ten unit system for the given scheduled time duration which is divided into 24 intervals. The ten unit test system data with non-smooth fuel cost and emission functions is taken from [11]. The load demand for 24 intervals is taken from [11]. From the simulations it is observed that solutions satisfy the constraints load demand plus transmission loss in each interval and also dynamic constraint ramp-rate constraint. The DEED optimization problem is also solved with PSO algorithm separately for estimating the performance of the proposed BPSO-DE algorithm. The proposed methods are coded in MATLAB 7.4 and run on Pentium (R) 4 CPU, 2.99GHz, 1GB RAM.

The parameters chosen for this system are $N_i = 125$, $N_{ad} = 2$, $N_{re} = 2$, $N_j = 4$, $N_b = 60$ and $P_{ad} = 0.2$. 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 $C_1 = 1.2, C_2 = 0.1$ to 0.5, $CR = 0.5$ to 0.8, $sf = 0.1$ to 0.3 and
$k_e=0.1$ or $0.2$, the convergence leads to better cost and emission.

The weighting factor $w$ is varied such that the emission objective is considered when $w=0$ and cost objective is considered when $w=1$. At $w=0.5$ equal importance is given to both cost and emission objectives. The DEED problem is solved for three cases. In case 1 the DEED problem is solved without considering power losses and valve point loading effects. In case 2 the problem is solved by considering both power losses and valve point loading effects. The parameter settings are same for all the cases. In each case the DEED problem is solved by a new hybrid method BPSO-DE and for comparison purpose the problem is also solved by PSO algorithm.

A. Case 1 and Case 2

In these cases the DEED problem without/with power losses and valve point loading effects are solved by BPSO-DE algorithm for case 1 and case 2 respectively. When $w=0$ the cost is 2495385.066 $ and emission is 244991.060 lb and the time taken per iteration is 4.34 seconds. The best scheduling of real powers of the generator for 24 hours dispatch interval when $w=0$ are shown in Fig.1. When $w=1$ the cost is 2309069.752 $ and emission is 276567.382 lb and the time taken per iteration is 4.54 seconds. The best scheduling when $w=1$ is shown in Fig. 2. When $w=0.5$ the cost is 2368338.714 $, emission is 255586.836 lb and the time taken per iteration is 4.26 seconds. The best scheduling when $w=0.5$ is given in Fig.3. Table 1 compares the cost and emission for the methods BPSO-DE and PSO at different weighting factors ($w=0$, $w=1$ and $w=0.5$) and those results reported in the literature for Case 2. It is observed from the results in Table 1 that at $w=1$ the cost obtained through BPSO-DE algorithm is less than the cost obtained through PSO algorithm. At $w=0$ the emission obtained through BPSO-DE algorithm is less than the emission obtained through PSO algorithm. Finally at $w=0.5$ both emission and cost obtained by BPSO-DE algorithm is less than that of PSO algorithm and other methods reported in the literature.

X. CONCLUSIONS

The paper presented a new hybrid approach namely BPSO-DE algorithm for solving the DEED problem with non linear constraints. In the proposed BPSO-DE method BFOA, PSO and DE techniques were integrated to form a new hybrid method. This integration helps to start with a good solution initially and the to get good final solution. The BFOA performs local search through chemo-tactic movements and the global search is accomplished over entire search space by PSO and DE operators. Simulations were carried out on a ten unit test system. The DEED problem was solved for three different cases for estimating the performance of the proposed BPSO-DE method. The comparative study showed that the superiority and feasibility of proposed BPSO-DE technique to solve the DEED problem on the other methods reported in the literature.
COST COMPARISON AT DIFFERENT WEIGHTING FACTORS FOR CASE 2

<table>
<thead>
<tr>
<th>Weight</th>
<th>Method</th>
<th>Cost ($)</th>
<th>Emission (lb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>W=0</td>
<td>BPSO-DE</td>
<td>2571386,388</td>
<td>29184,844</td>
</tr>
<tr>
<td></td>
<td>PSO</td>
<td>2577794,261</td>
<td>296147,273</td>
</tr>
<tr>
<td></td>
<td>RC GA [11]</td>
<td>2656300,000</td>
<td>304120,000</td>
</tr>
<tr>
<td>W=0.5</td>
<td>BPSO-DE</td>
<td>2519329,410</td>
<td>304382,692</td>
</tr>
<tr>
<td></td>
<td>PSO</td>
<td>2520415,354</td>
<td>305027,288</td>
</tr>
<tr>
<td></td>
<td>RC GA [11]</td>
<td>2521000,000</td>
<td>312460,000</td>
</tr>
<tr>
<td>W=1</td>
<td>NSGA-II [11]</td>
<td>2522600,000</td>
<td>309940,000</td>
</tr>
<tr>
<td></td>
<td>BPSO-DE</td>
<td>2487882,699</td>
<td>329000,796</td>
</tr>
<tr>
<td></td>
<td>PSO</td>
<td>2489782,072</td>
<td>306650,969</td>
</tr>
<tr>
<td></td>
<td>RC GA [11]</td>
<td>2516800,000</td>
<td>317400,000</td>
</tr>
</tbody>
</table>

XI. REFERENCES


