01 Sep 2008

Economic Load Dispatch Using Bacterial Foraging Technique with Particle Swarm Optimization Biased Evolution

Ahmed Yousuf Saber

Ganesh K. Venayagamoorthy
Missouri University of Science and Technology

Recommended Citation

Economic Load Dispatch using Bacterial Foraging Technique with Particle Swarm Optimization Biased Evolution

Ahmed Y. Saber

Dept. of Electrical and Computer Engineering

King Abdulaziz University, P.O. Box 80204

Jeddah 21589, KSA

aysaber@ieee.org

Ganesh K. Venayagamoorthy

Real-Time Power and Intelligent Systems Laboratory

Missouri University of Science and Technology

MO 65409-0040, USA

gkumar@ieee.org

Abstract—This paper presents a novel modified bacterial foraging technique (BFT) to solve economic load dispatch (ELD) problems. BFT is already used for optimization problems, and performance of basic BFT for small problems with moderate dimension and searching space is satisfactory. Search space and complexity grow exponentially in scalable ELD problems, and the basic BFT is not suitable to solve the high dimensional ELD problems, as cells move randomly in basic BFT, and swarming is not sufficiently achieved by cell-to-cell attraction and repelling effects for ELD. However, chemotaxis, swimming, reproduction and elimination-dispersal steps of BFT are very promising. On the other hand, particles move toward promising locations depending on best values from memory and knowledge in particle swarm optimization (PSO). Therefore, best cell (or particle) biased velocity (vector) is added to the random velocity of BFT to reduce randomness in movement (evolution) and to increase swarming in the proposed method to solve ELD. Finally, a data set from a benchmark system is used to show the effectiveness of the proposed method and the results are compared with other methods.

Index Terms—Bacterial foraging technique, particle swarm optimization, economic load dispatch.

I. Introduction

Load demand is distributed among running units in economic load dispatch (ELD) problems. An optimal ELD should meet load demand, generation limit, ramp rate, prohibited operating zone, etc. considering network losses at every time interval such that the total cost is minimum [1]. It is the main complex computational intensive part of unit commitment problems [2]. Therefore, ELD is one of the most important problems in power systems optimization area.

A bibliographical survey on ELD methods reveals that various numerical optimization techniques have been employed to approach the ELD problem. ELD is solved traditionally using mathematical programming based on optimization techniques such as lambda iteration, gradient method, dynamic programming (DP) and so on [1, 3-6].

Complex constrained ELD is addressed by intelligent methods. Among these methods, some of them are genetic algorithm (GA) [7-8], evolutionary programming (EP) [9-10], tabu search [11], hybrid EP [12], neural network (NN) [13], adaptive Hopfield neural network (AHNN) [14], particle swarm

optimization (PSO) [15-23]. Basic bacterial foraging technique (BFT) is described in [24-26], and BFT is applied for optimization in [27]; however, researchers have not tried to solve ELD using BFT.

Lambda iteration, gradient method [1, 3-4] can solve simple ELD calculations, which are not sufficient for real applications. However, they are fast. Intelligent methods in [7-27] are general and thus they have randomness/blindness for a particular problem. For complex ELD problems, the intelligent methods should be modified so that they can solve economic dispatch problems properly and efficiently.

Some general optimization software packages are also available for modeling ELD problems [28-29]. However, software companies urge state-of-the-art technologies to continuously update their products.

Swarm optimization methods are very popular in recent days because they have information sharing and conveying mechanisms. Among swarm optimization methods, bacterial foraging and PSO are very promising. Each method has different set of advantages and disadvantages regarding local minima, randomness, direction of movement, attraction/repelling, swarming and so on. Random velocity of BFT is improved by using PSO movement (evolution). Best cell (particle) biased velocity (vector) is applied in the proposed method in addition to use the random velocity of BFT - called BFT with PSO biased evolution (BFT-PSOBE). The proposed method therefore includes advantages of both bacterial foraging technique and particle swarm optimization, and also excludes disadvantages of bacterial foraging technique.

BFT with PSO biased evolution for ELD is introduced first time in this paper and the rest of the paper is organized as follows. In Section II, problem formulation and constraints of ELD are discussed. The proposed method, applied distributions, algorithm and constraints management are explained in Section III. Simulation results are reported in Section IV. Finally, conclusion is drawn in Section V.

II. ELD Problem Formulation

A. Nomenclature and Acronyms

The following notations are used in this paper.

N	Number of units of a system
p_i	Output power of ith unit
$p_{i,m}$	Output power of unit m for bacterium i
P_i	Position of bacterium i
P_i^{max}	Maximum output limit of ith unit
P_i^{min}	Minimum output limit of ith unit
$P_i^{max}(t)$	Maximum output power of unit i at time t considering ramp rate
$P_i^{min}(t)$	Minimum output power of unit i at time t considering ramp rate
D	Demand
R	System reserve
$\mathcal{FC}()$	Fuel cost function
$\mathcal{TC}$	Total cost
RUR_i	Ramp up rate of unit i
RDR_i	Ramp down rate of unit i
P_{loss}	Network losses
$J()$	Objective/cost function

B. Objective Function

The objective of ELD problem is the minimization of total generation cost considering equality and inequality constraints. Fuel cost function:

Main generation cost is fuel cost. Usually, fuel cost of a thermal unit is expressed as a second order approximate function of its output p_i.

$$\mathcal{FC}_i(p_i) = a_i + b_i p_i + c_i p_i^2 \qquad (1)$$

where a_i, b_i and c_i are positive fuel cost coefficients of unit i.

To take care about valve-point effects, sinusoidal functions are added as below.

$$\mathcal{FC}_i(p_i) = a_i + b_i p_i + c_i p_i^2 + |e_i \sin(f_i(P_i^{min} - p_i))| \qquad (2)$$

where e_i and f_i are valve-point coefficients of unit i, and the cost function is a non-convex complex polynomial.

Therefore, the objective function of ELD for an N-unit system is

$$\min_{p_i} \mathcal{TC} = J(p_1, p_2, \ldots, p_N) = \sum_{i=1}^{N} \mathcal{FC}_i(p_i) \qquad (3)$$

subject to the following constraints.

In other words, the goal is to determine p_i, $i = 1, 2, \ldots, N$, so that $\mathcal{TC}$ is minimum subject to the following constraints in Section II-C. Any new type of cost may be included (e.g., carrying cost, maintenance cost, emission cost and so on) or any existing type of cost may be excluded from the objective function according to the system operators' demand. Different weights may also be assigned to different types of cost depending on their relative importance in the changing environment.

C. Constraints

The constraints that must be satisfied during ELD optimization process are as follows.

1. System power balance
Total generated power from all committed units must satisfy load demand.

$$D = \sum_{i=1}^{N} p_i - P_{loss}. \qquad (4)$$

Power loss (P_{loss}), which depends on physical geographical network and generated power level (p_i), should be minimum for the generation cost minimization.

2. Spinning reserve
Adequate spinning reserves are required to increase reliability of the system.

$$\sum_{i=1}^{N} P_i^{max} \geq D + P_{loss} + R. \qquad (5)$$

Fixed amount or a predefined percentage (e.g., 5%) of maximum load demand is used as spinning reserve. Higher spinning reserve makes a system more reliable; however, it will increase running cost. Usually spinning reserve is fulfilled in unit commitment scheduling.

3. Generation limits
Each unit has generation range, which is represented as

$$P_i^{min} \leq p_i \leq P_i^{max}. \qquad (6)$$

This constraint prohibits a cheap unit to generate power more than its maximum limit as well as an expensive unit to generate power less than its minimum limit.

4. Ramp rate
For each unit, output is limited by time dependent ramp up/down rate at each hour as given below.

$$P_i^{min}(t) \leq p_i(t) \leq P_i^{max}(t) \qquad (7)$$

where $P_i^{min}(t) = \max\ (p_i(t-1) - RDR_i,\ P_i^{min})$ and $P_i^{max}(t) = \min\ (p_i(t-1) + RUR_i,\ P_i^{max})$.

5. Prohibited operating zones
In practical operations, the generated output p_i of unit i must avoid operations in prohibited zones. The feasible operating zones of unit i can be described as

$$p_i \in \begin{cases} P_i^{min} \leq p_i \leq p_{i,1}^l \\ p_{i,j-1}^u \leq p_i \leq p_{i,j}^l, & j = 2, 3, \ldots, Z_i \\ p_{i,Z_i}^u \leq p_i \leq P_i^{max} \end{cases} \qquad (8)$$

where $p_{i,j}^l$ and $p_{i,j}^u$ are lower and upper bounds of the jth prohibited zone of unit i, and Z_i is the number of prohibited zones of unit i.

6. Network losses
In the economic dispatch, network losses are taken into account as functions of generated outputs and B coefficients matrix [1, 8].

$$P_{loss} = \sum_{i=1}^{N} \sum_{j=1}^{N} p_i B_{ij} p_j + \sum_{i=1}^{N} B_{i0} p_i + B_{00}. \qquad (9)$$

[B_{ij} = ijth element of loss coefficient symmetric matrix B

B_{i0} = ith element of the loss coefficient vector

B_{00} = loss coefficient constant]

7. Initial status

Initial status must be considered, as ELD is a part of unit commitment problem [1-2].

III. PROPOSED METHOD

A. Bacterial Foraging for Optimization

To solve non-gradient optimization problems such as minimization, where analytical description of gradient is not present, bacterial foraging (BF) is one of the suitable methods. Let P_i be the initial position of bacterium i, and $J(P_i)$ represents an objective function with the combined effect of attractants and repellents from the environment. Let $J(P_i) < 0$, $J(P_i) = 0$ and $J(P_i) > 0$ represent the bacterium at location P_i is in nutrient rich, neutral and noxious environments, respectively. Chemotaxis is a foraging behavior that implements a type of optimization where bacteria try to climb up the nutrient concentration (i.e., lower and lower values of $J(P_i)$) and avoid being at positions P_i where $J(P_i) \geq 0$ [26].

With the initial position P_i, a bacterium (i) takes a chemotactic step (j) with the step size $C(i)$ and calculates the objective function value ($J(P_i)$) at each step. If at position $P_i(j + 1)$, the value J is better than at position $P_i(j)$, then another step of same size $C(i)$ in this same direction will be taken again and again, if that step resulted in a position with a better value than at the previous step. This is a swim step and is continued until minimum value is reached but only up to a maximum number of steps (N_s). After N_c chemotactic steps, a reproduction step (N_{re}) is taken in which the population is sorted in ascending order of the objective function value (J) and the least healthy bacteria are replaced by copies of the healthiest bacteria. The reproduction step is followed by elimination-dispersal (N_{ed}) event. For each elimination-dispersal event, each bacterium in the population is subjected to elimination-dispersal (i.e., to eliminate a bacterium simply disperse one to a random location on the optimization domain) with probability p_{ed}.

Therefore, basically the bacterial foraging technique implements a type of biased random walk, which is not suitable for ELD problems in huge multi-dimensional space with constraints. This randomness is decreased by using PSO.

B. Overview of PSO Algorithm

PSO is similar to other evolutionary algorithms where the system is initialized with a population of random solutions. Each potential solution, called a particle, flies in N-dimensional problem space with a velocity, which is dynamically adjusted according to the flying experiences of its own and the best particle in the swarm [20-22].

$$v_{ij} = w * v_{ij} + c_1 * rand()(pbest_{ij} - p_{ij}) +$$
$$c_2 * rand()(gbest_j - p_{ij}). \quad (10)$$
$$p_{ij} = p_{ij} + v_{ij}. \quad (11)$$

The location of the ith particle is represented as $P_i = [p_{i1}, p_{i2}, \ldots, p_{iN}]^T$. The best previous position of the ith particle is recorded as $pbest_i$. The index of the best $pbest$ among all the particles is represented by the symbol g. The location $pbest_g$ is called $gbest$. The velocity of ith particle is represented as $V_i = [v_{i1}, v_{i2}, \ldots, v_{iN}]^T$. Next velocity and position of each particle are calculated using current velocity and the distances from $pbest$, $gbest$. In velocity calculation, the first term indicates the current velocity of the particle (inertia), second term presents the cognitive part of PSO where the particle changes its velocity based on its own thinking and memory, and the third term is the social part of PSO where the particle changes its velocity based on the social-psychological adaptation of knowledge. All the terms are multiplied by appropriate parameters.

C. Data Structure of the Proposed Method

In this paper, the position of ith bacterium in an N-dimensional space is represented by P_i (e.g., $P_i = [p_{i,1}, p_{i,2}, \ldots, p_{i,N}]^T$) where N is the number of units of the system and $p_{i,m}$ is the output power of mth unit for ELD. Swarm of bacteria is denoted by $\mathcal{P}$, i.e., $\mathcal{P} = \bigcup P_i = \{P_1, P_2, \ldots, P_S\}$ where $|\mathcal{P}| = S$ is total number of bacteria in the population. Let j, k and l be the indices of chemotactic step, reproduction step and elimination-dispersal event, respectively; and they are used as superscripts in the paper. Therefore, $P_i^{j,k,l} = [p_{i,1}^{j,k,l}, p_{i,2}^{j,k,l}, \ldots, p_{i,N}^{j,k,l}]^T$ is the position of bacterium i at the jth chemotactic step, kth reproduction step and lth elimination-dispersal event.

Data structure and efficient coding are important for time complexity of an algorithm. Bacterial foraging technique has mainly 3 nested loops - chemotactic, reproduction and elimination-dispersal loops. Basic structure of BFT is shown below.

Elimination-dispersal loop
{ Reproduction loop
    { Chemotactic loop
        { Tumble, move, swim, calculate cell-to-cell attraction
            and repelling effects, fitness, etc.
        }
    Reproduction operations;
    }
  Elimination-dispersal operations;
}

D. The Optimization Function of BFT with PSO Biased Evolution

The main objective of this paper is to propose a modified version of BFT, which is suitable for ELD problems in huge multi-dimensional space. Therefore, PSO biased evolution is included to reduce disadvantages of BFT in the proposed method.

1) Chemotaxis: In BFT, random velocity is used in chemotaxis operations. Therefore, it is one kind of blind search and it does not converge quickly. Besides, swarming effect is not satisfactory for the ELD problem using cell-to-cell attraction and repelling effects in BFT, as the ELD is a complex problem

in huge multi-dimensional space with constraints. Therefore, the global best location (*gbest* particle of PSO) directed/biased velocity vector is added with the random velocity in this research. In (12), first term (random velocity) helps to avoid local minima, and second term (global best biased velocity) decreases randomness and increases swarming effect in the proposed method.

Velocity and swarming:

$$v_{i,m}^{j,k,l} = \text{random}(-1,1) + \\ c_2 * \text{random}(0,1)(gbest_m^{j,k,l} - p_{i,m}^{j,k,l}). \quad (12)$$

Unit vector:

$$\hat{u}_{i,m}^{j,k,l} = \frac{v_{i,m}^{j,k,l}}{\|V_i^{j,k,l}\|} = \frac{v_{i,m}^{j,k,l}}{\sqrt{\sum_{m=1}^{N}(v_{i,m}^{j,k,l})^2}}. \quad (13)$$

New position and swimming:

$$C_i = 2.5 + 0.1 * i. \quad (14)$$

$$p_{i,m}^{j+1,k,l} = p_{i,m}^{j,k,l} + C_i * \hat{u}_{i,m}^{j,k,l}. \quad (15)$$

Where $V_i^{j,k,l}$ is the modified PSO directed velocity and the normalized vector $\hat{U}_i^{j,k,l} = [\hat{u}_{i,1}, \hat{u}_{i,2}, \ldots, \hat{u}_{i,N}]^T$ of a non-zero velocity $V_i^{j,k,l} = [\hat{v}_{i,1}, \hat{v}_{i,2}, \ldots, \hat{v}_{i,N}]^T$ is a unit vector co-directional with $V_i^{j,k,l}$, which indicates direction of next movement. $\|V_i^{j,k,l}\|$ is the norm (or length) of $V_i^{j,k,l}$. Unit vector is sometimes used as a synonym of the normalized vector in this paper. For the $v_{i,m}^{j,k,l}$ calculation, first term and second term come from BFT and PSO, respectively. As $gbest_m^{j,k,l}$ is the best cell (particle in PSO) in the swarm, the proposed method has more directed movement and swarming effect than standard BFT. $\hat{u}_{i,m}^{j,k,l}$ is current velocity of mth available unit among total N units, C_i is linear increasing step size (already sorted) and $p_{i,m}^{j+1,k,l}$ is the new position after PSO biased evolution using the above mentioned velocity, direction and step length from previous position $p_{i,m}^{j,k,l}$. Considering ELD problem, if output level $p_{i,m}^{j+1,k,l}$ is outside of valid generation range or inside of prohibited zones of unit m, then the nearest valid output level of unit m is assigned to $p_{i,m}^{j+1,k,l}$ directly - called direct repair. The proposed linear increasing step size (C_i) helps to get better balance between local and global searching abilities. Minimum value of C_i is 2.6 (=2.5+0.1) from prior simulation. If the new position is better than previous position, then another step of same size C_i in this same direction will be taken again and again, if that step resulted in a position with a better value than at the previous step. This is a swim step and is continued until minimum value is reached but only up to a maximum number of steps (N_s). This represents that a cell will tend to keep moving if it is headed in the direction of increasingly favorable environments [26].

Swarming:

$$J_{cc}(P_i, \mathcal{P}) = \sum_{t=1}^{|\mathcal{P}|}[-d_{attract}\exp(-w_{attract}(\|\bar{P}_i - \bar{P}_t\|)^2)] \\ + \sum_{t=1}^{|\mathcal{P}|}[h_{repellent}\exp(-w_{repellent}(\|\bar{P}_i - \bar{P}_t\|)^2)] \\ = \sum_{t=1}^{|\mathcal{P}|}[-d_{attract}\exp(-w_{attract}\sum_{m=1}^{N}(p_{i,m} - p_{t,m})^2)] \\ + \sum_{t=1}^{|\mathcal{P}|}[h_{repellent}\exp(-w_{repellent}\sum_{m=1}^{N}(p_{i,m} - p_{t,m})^2)] \quad (16)$$

where $J_{cc}(P_i, \mathcal{P})$ denotes the combined cell-to-cell attraction and repelling effects for bacterium i at position $P_i = [p_{i,1}, p_{i,2}, \ldots, p_{i,N}]^T$ and whole swarm of bacteria $\mathcal{P} = \{P_1, P_2, \ldots, P_S\}$. Mathematically $J_{cc}()$ is an exponential function, and its value initially decreases and then increases. This cell-to-cell signaling helps cells to move toward other cells, but not too close to them [26]. $h_{repellent}$, $w_{repellent}$ are height, width of the repellent and $d_{attract}$, $w_{attract}$ are depth, width of the attractant, respectively. Therefore, the proposed method incorporates double swarming effect from PSO biased evolution and cell-to-cell signaling.

Fitness function:

$$fit_i = \sum_{m=1}^{N}\mathcal{FC}_m(p_{i,m}^{j+1,k,l}) + J_{cc}(P_i, \mathcal{P}) + Penalty. \quad (17)$$

$$Penalty = \frac{\mathcal{TC}}{\sum_{m=1}^{N}p_{i,m} - P_{loss}}(Error_i)^2. \quad (18)$$

$$Error_i = \sum_{m=1}^{N}p_{i,m} - P_{loss} - D. \quad (19)$$

Where fit_i is the fitness of cell i. As ELD is a minimization problem, lower fitness value of fit_i represents better cell/particle i. From the least fitness value, the best cell/particle (*gbest*) is selected for the velocity with PSO biased evolution. In the fitness function, third term is penalty for constraints and it is proportional to the square of error, $Error_i$ for bacterium i. For an equality constraint (e.g., system power balance), the error is the difference between its left and right sides. Penalty is zero when constraints are fulfilled and in that case, fitness value will be small. If the current fitness of bacterium i is better than the fitness of current best cell, then $gbest_m^{j+1,k,l}$ will be updated by $p_{i,m}^{j+1,k,l}$ for all dimensions. As *gbest* is selected from the best cell (least fitness value) and invalid locations are discouraged, the system converges to the location with satisfying constraints gradually.

Typically, $\mathcal{TC}$ is a second order polynomial for ELD. However, the system can handle higher order cost polynomials for the $\mathcal{TC}$ with no extra difficulty, as no extra new method or different equation is needed to handle these higher order cost polynomials, except some extra similar calculations.

TABLE I
STRENGTH OF THE PROPOSED METHOD WITH RESPECT TO
INDIVIDUAL PSO AND BFT

	PSO	BFT	Proposed BFT with PSO bi-ased evolution
Velocity	Directed	Random	Random plus PSO directed
Swarming effect	Yes	Yes	Yes
Attractant	Yes	Yes	Yes
Repellent	No	Yes	Yes
Unit vector	No	Yes	Yes
Step length	No	Yes (fixed)	Yes (variable)
Swimming in the same direction	No	Yes	Yes

2) Reproduction: After N_c chemotactic steps, a reproduction step is executed. In this step, the weakest $S_r = S/2$ bacteria die and the rest S_r healthiest bacteria each split into two bacteria, which are placed at the same location. This ensures that the population of bacteria remains constant. A popular convention is to consider S_r as 50% of S where S is a positive even integer. It reduces unpromising diversity in the searching space to accelerate the process [26].

3) Elimination-dispersal event: Let N_{ed} be the number of elimination-dispersal events. In elimination-dispersal steps, bacteria in a region may be destroyed or may disperse to a new region full of good nutrients. Each bacterium is subjected to elimination-dispersal with a probability of p_{ed}, where the bacterium may be dispersed into an unexplored region of environment or searching space. While this may destroy the progress achieved through the chemotactic process thus far, it may happen that the bacterium may find itself closer to new source of nutrients. It increases global searching ability. After every N_c chemotactic steps are completed, one reproduction step is undertaken and after N_{re} reproduction steps are completed, one elimination-dispersal step is undertaken. A typical relationship among N_c, N_{re} and N_{ed} is $N_c > N_{re} > N_{ed}$. Hence a bacterium will undergo several chemotactic steps before it is allowed to reproduce and the system will undergo several generations before an elimination-dispersal step is executed. Strength of the proposed BFT with PSO biased evolution is shown in Table I with respect to individual PSO and BFT. Algorithm of the proposed method is shown below.

```
 1) Initialization: Generate locations of bacteria
    randomly; assign large value for fitness of each
    bacterium ft; assign random gbest and large fitness
    value global_ft initially; initialize N, S, Nc, Nre,
    Ned, ped, Ns, dattract, wattract, hrepellent, wrepellent and
    so on.
 2) Elimination-dispersal loop: l=l+1 (start from l = 0)
    //Outer loop
    3) Reproduction loop: k=k+1 (start from k=0) //Middle
       loop
       4) Chemotaxis loop: j=j+1 (start from j=0) //Inner loop
       4.1) For i=1,2,...,S, take a chemotactic step for
            bacterium i as follows.
       4.2) Velocity with PSO biased evolution:
```
$$v_{i,m} = \text{random}(-1,1) + \text{random}(0,1)(gbest_m - p_{i,m}),$$
where m=1,2,...,N for all dimensions.
```
       4.3) Unit vector:
```
$$\hat{u}_{i,m} = \frac{v_{i,m}}{\sqrt{\sum_{m=1}^{N}(v_{i,m})^2}}, \text{ where } m=1,2,...,N \text{ for all dimensions.}$$

```
       4.4) Move to new position:
```
$C_i = 2.5 + 0.1 * i$ // *Linear increasing step size*
$p_{i,m} = p_{i,m} + C_i * \hat{u}_{i,m}$, where m=1,2,...,N for all dimensions.
```
       4.5) Handle constraints:
            (a) Generation limits:
```
If $p_{i,m} > P_i^{max}$ then $p_{i,m} = P_i^{max}$ and
If $p_{i,m} < P_i^{min}$ then $p_{i,m} = P_i^{min}$,
where m=1,2,...,N for all dimensions.
```
            (b) Ramp rate and prohibited zones:
                Similar statements of Step 4.5(a).
       4.6) Swarming and fitness:
            i)
```
$J_{cc}(P_i, \mathcal{P}) =$
$$\sum_{t=1}^{S}[-d_{attract}\exp(-w_{attract}\sum_{m=1}^{N}(p_{i,m} - p_{t,m})^2)] +$$
$$\sum_{t=1}^{S}[h_{repellent}\exp(-w_{repellent}\sum_{m=1}^{N}(p_{i,m} - p_{t,m})^2)];$$
```
            ii)
```
$fit_i(P_i) = \sum_{m=1}^{N}\mathcal{FC}_m(p_{i,m}) + J_{cc}(P_i, \mathcal{P}) + Penalty$
```
       4.7) Swimming:
            Swim_count = 0
            While (fiti(Pi) < fti) and (Swim_count < Ns)
            i) Swim_count = Swim_count + 1
            ii) fti = fiti(Pi)
            iii) pi,m = pi,m + Ci * ûi,m; m=1,2,...,N
       4.8) Update best location for PSO biased
            evolution:
            If fti < global_ft then gbestm=pi,m;
            m=1,2,...,N
       4.9) Next bacterium i=i+1 and continue operations
            in steps 4.2)-4.8) for all the bacteria.
       4.10) If j<Nc then go to Step 4).
    3.1) Reproduction operations:
         Sort bacteria in order of ascending fitness fti.
         The weakest Sr = S/2 bacteria die and the rest
         Sr best bacteria each split into two bacteria,
         which are placed at the same location.
    3.2) If k<Nre then go to Step 3).
 2.1) Elimination-dispersal operation:
      For each bacterium
      If random(0,1)<ped then pi,m = Pm^min +
      random(Pm^max - Pm^min), m=1,2,...,N
 2.2) Stopping criterion:
      If l < Ned then go to Step 2); otherwise print
      results and end.
```

E. Constraints Management

Constraints management is one of the major issues of optimization. As there is a set of physical and operational constraints inherent in the ELD problem, the generated power level of a new location may not satisfy all the constraints. In the proposed method, direct repair and penalty are applied to handle constraints. Generation limit, ramp rates and prohibited zones are straight forward constraints, and they are directly managed by assigning the nearest valid generation level immediately if any limit of a unit is violated - called direct repair. System power balance, including network losses, is handled by penalty. It is already mentioned that $Error$ is the difference between its left and right sides of system power balance equation and the system converges to the location with satisfying the constraint gradually. This penalty can also handle transmission constraint; however, this constraint is not included in this paper. If spinning reserve is violated at any scheduling period, the system suffers from deficiency in units. Then, available decommitted units are forced to turn on randomly until it is satisfied. Spinning reserve is an important constraint for reliable operations and it is usually fulfilled in unit commitment scheduling.

Unit name	a_i ($)	b_i ($/MW)	c_i ($/MW^2)	P_i^{max} (MW)	P_i^{min} (MW)	P_i^0 (MW)	RUR_i (MW/h)	RDR_i (MW/h)	Prohibited zones (MW)
Unit 1	240	7.0	0.0070	500	100	440	80	120	[210 240][350 380]
Unit 2	200	10.0	0.0095	200	50	170	50	90	[90 110][140 160]
Unit 3	220	8.5	0.0090	300	80	200	65	100	[150 170][210 240]
Unit 4	200	11.0	0.0090	150	50	150	50	90	[80 90][110 120]
Unit 5	220	10.5	0.0080	200	50	190	50	90	[90 110][140 150]
Unit 6	190	12.0	0.0075	120	50	110	50	90	[75 85][100 105]

Method	Success (%)	Total cost				Execution time		
		Best ($)	Worst ($)	Average ($)	Variation (%)	Max. (sec)	Min. (sec)	Average (sec)
Proposed BFT with PSO	100	15,439.45	15,441.26	15,439.86	0.0117	6.92	4.83	6.12
BFT	100	15,455.65	15,489.61	15,466.30	0.2197	5.98	4.46	5.32
PSO[15]	-	15,450.00	15,492.00	15,454.00	0.27	-	-	14.89
GA	-	15,459.00	15,524.00	15,469.00	0.42	-	-	41.58

F. Stopping Criterion

An iterative method is stopped running when there is no significant improvement in the solution or the maximum number of iterations is reached. In this study, the stopping criterion is the maximum number of iterations, i.e., N_{ed} elimination-dispersal iterations, N_{re} reproduction iterations for each elimination-dispersal step and N_c chemotactic iterations for each reproduction step. It is already mentioned that the typical relationship among N_c, N_{re} and N_{ed} is $N_c > N_{re} > N_{ed}$. Depending on the problem size and constraints, values of N_c, N_{re}, N_{ed} and S should be chosen. Very large values of the above parameters spend much longer execution time with a slight improvement of solution quality.

IV. SIMULATION RESULTS

All calculations have been run on Intel(R) Celeron(TM) 2.60 GHz CPU, 1 GB RAM, Windows XP OS and C/C++ compiler. One standard data set (e.g., 6-unit system) is used to compare with other popular methods. Attributes of the proposed BFT with PSO biased evolution by trial and error for ELD are as follows:

S = 10, N_c = 40, N_{re} = 15, N_{ed} = 10, p_{ed} = 0.25, N_s = 10, $d_{attract}$ = 1000, $w_{attract}$ = 0.0020, $h_{repellent}$ = 1000, $w_{repellent}$ = 0.0100, and $c2$ = 2.5.

Input data are collected from 6-unit system. The system consists of 26 buses and 46 transmission lines. Input data are shown in Table II and B coefficients (base capacity 100 MVA) for network losses are given below [15].

$$B_{ij} = \begin{vmatrix} 0.0017 & 0.0012 & 0.0007 & -0.0001 & -0.0005 & -0.0002 \\ 0.0012 & 0.0014 & 0.0009 & 0.0001 & -0.0006 & -0.0001 \\ 0.0007 & 0.0009 & 0.0031 & 0.0000 & -0.0010 & -0.0006 \\ -0.0001 & 0.0001 & 0.0000 & 0.0024 & -0.0006 & -0.0008 \\ -0.0005 & -0.0006 & -0.0010 & -0.0006 & 0.0129 & -0.0002 \\ -0.0002 & -0.0001 & -0.0006 & -0.0008 & -0.0002 & 0.0150 \end{vmatrix},$$

$B_{i0} = 1.0e^{-3}[-0.3908 \quad -0.1297 \quad 0.7047 \quad 0.0591 \quad 0.2161 \quad -0.6635]$,
$$B_{00} = 0.0056.$$

Test results using different methods are shown in Table III. The best, worst and average findings of the proposed method

are reported together with cost variation as a percentage of the best solution. Constraints, including ramp rate, network losses, prohibited zones, are considered here. It always converges and variation is tolerable. For more than sufficient iterations, best, worst and average results are near about the same and the variation is negligible. Average cost and execution time of 10 runs are near to the best result. These facts strongly demonstrate the robustness of the proposed BFT with PSO biased evolution for the ELD problem. Percentage of success, maximum execution time and minimum execution time were not reported for GA and PSO in [15].

Fig. 1 shows the movement of the best bacterium visually using the proposed BFT with PSO biased evolution. Initially, values are frequently changed in all dimensions (6-dimension as 6-unit system); however, the change is relatively smaller near to the final generations. It indicates a fine tuning of the searching space. Table IV shows the comparison of the proposed method to popular methods (e.g., GA and PSO reported in [15], new PSO and local random search (NPSO-LRS) reported in [18], and standard BFT alone) considering the generated best output. According to Tables III and IV, the proposed BFT with PSO biased evolution provides the lowest cost and fastest schedule where all the constraints are fulfilled.

Fig. 2 shows swarm behavior of all the bacteria. Data are collected after end of each elimination-dispersal loop and the
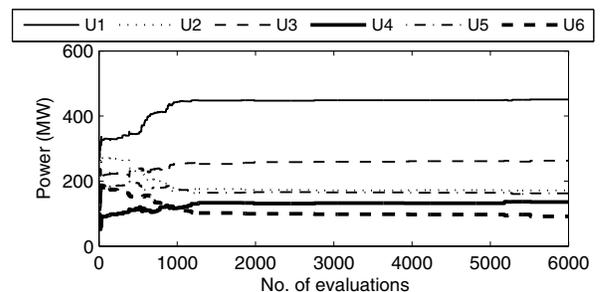

Fig. 1. Movement of the best bacterium (6-unit system).

TABLE IV
COMPARISON OF BEST OUTPUTS OF 6-UNIT SYSTEM USING DIFFERENT METHODS

	GA method[15]	PSO method[15]	NPSO-LRS method[18]	BFT method	Proposed BFT with PSO method	P_i - P_i^0	Inside/outside of prohibited zones
Unit 1 (MW)	474.807	447.497	446.960	448.688	450.129	10.129	Outside
Unit 2 (MW)	178.636	173.322	173.394	164.007	173.623	3.623	Outside
Unit 3 (MW)	262.208	263.474	262.344	263.156	260.607	60.607	Outside
Unit 4 (MW)	134.282	139.059	139.512	131.657	139.489	-10.511	Outside
Unit 5 (MW)	151.903	165.476	164.709	197.039	159.697	-30.303	Outside
Unit 6 (MW)	74.181	87.128	89.016	71.488	91.507	-18.493	Outside
Network loss (MW)	13.0217	12.9584	12.936	13.0054	12.0525	-	-
Total generated power (MW)	1276.0300	1276.0100	1275.94	1276.0339	1275.0524	-	-
Load demand (MW)	1263.00	1263.00	1263.00	1263.00	1263.00	-	-
Error (MW)	0.0083	0.0516	0.004	0.0285	0.0001	-	-
Cost ($)	15,459.00	15,450.00	15,450.00	15,455.65	15,439.45	-	-

Note: '-' indicates not applicable

graph is plotted for 8 sets of data for simplicity. Initially they are random and then they are concentrating together gradually.

The proposed BFT with PSO biased evolution is superior to other mentioned methods because (a) the proposed method shares many common parts of GA; however, the proposed hybrid method, which consists of BFT and PSO, has better information sharing and conveying mechanisms than GA; (b) the proposed method incorporates both BFT and PSO effects in movement to converge quickly by making early jumps (with respect to PSO) from local minima; (c) it increases swarming, as it has double swarming effect; (d) it reduces randomness of standard BFT, as PSO biased evolution is included here; and (e) it has better balance between local and global searching abilities.

Execution time depends on algorithm, computer configuration and efficient program coding. It is already mentioned that the proposed method is less random than standard BFT, it has more swarming effect than BFT and PSO, and it converges very quickly. Besides, it satisfies constraints easily. The proposed method is implemented in C/C++ efficiently and run on a modern system. Therefore, the proposed method is the fastest among all the mentioned methods. This fact is reflected in Table III. It is promising and robust. Convergence of the proposed method is shown in Fig. 3.

Therefore, the above simulation results demonstrate the effectiveness of the proposed method in handing economic dispatch with all practical constraints. This method is practically applicable in unit commitment problems after real load forecasting, as its execution time is few seconds (for each hour dispatch calculations) and existing several-hour ahead load

forecasting methods are very popular in recent days. Besides, the proposed method is a generalized optimization method. Thus it can easily handle a new system of higher order cost functions where new constraints may be included or any existing constraint may be relaxed according to operators' demand.

V. CONCLUSION

This paper introduces modified BFT with PSO biased evolution for ELD to have better economic output levels of generators. In this study, the primary contribution is the appropriate introduction of BFT by incorporating the best bacterium in velocity to reduce randomness and to increase swarming effect. The proposed BFT has better information sharing and conveying mechanisms than other evolutionary methods. Advantages of the proposed method are discussed below.

- It is an improved version of standard BFT.
- It has less random property than standard BFT.
- It can handle constraints and higher order cost polynomials (both convex and non-convex) easily without extra concentration/effort.
- The proposed method is general and other methods (e.g., ELD for non-smooth cost functions, non-convex cost functions and so on) are just subsets of the proposed method.
- It does not need much memory.
- It is very fast.

To evaluate the proposed method properly, it should be mentioned that in this study, one more vector is added in velocity calculation. Thus it needs slightly more memory than standard BFT; however, it is quite acceptable for the solution
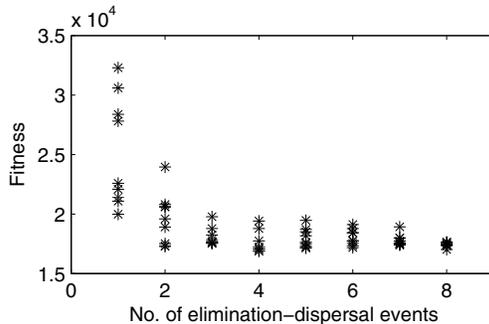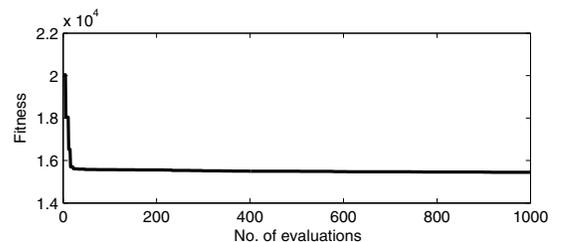

Fig. 2. Swarm behavior.

Fig. 3. Convergence of the proposed BFT with PSO biased evolution.

quality. Finally, this study is a first look at modified BFT with PSO biased evolution for ELD and there is enough scope to work on it for a real ELD application in unit commitment problems within practical execution time limit.

REFERENCES

[1] A. J. Wood and B. F. Wollenberg, "Power generation, operation, and control," 2nd ed., *New York: John Wiley and Sons*, 1996.

[2] A. Y. Saber, T. Senjyu, T. Miyagi, N. Urasaki, and T. Funabashi, "Fuzzy unit commitment scheduling using absolutely stochastic simulated annealing," *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 955-964, May 2006.

[3] P. Aravindhababu and K. R. Nayar, "Economic dispatch based on optimal lambda using radial basis function network," *Electrical Power and Energy Systems*, vol. 24, pp. 551-556, 2002.

[4] IEEE Committee Report, "Present practices in the economic operation of power systems," *IEEE Trans Power Appa. Syst.*, vol. PAS-90, pp. 1768-1775, 1971.

[5] B. H. Chowdhury and S. Rahman, "A review of recent advances in economic dispatch," *IEEE Trans Power Syst*, vol. 5, no. 4, pp. 1248-1259, 1990.

[6] J. A. Momoh, M.E. El-Hawary, and R. Adapa, "A review of selected optimal power flow literature to 1993, Part I: Nonlinear and quadratic programming approaches," *IEEE Trans Power Syst*, vol. 14, no. 1, pp. 96-104, 1999.

[7] D. C. Walters and G. B. Sheble, "Genetic algorithm solution of economic dispatch with valve point loading," *IEEE Trans. Power Syst.*, vol. 8, no. 3, pp. 1325-1332, Aug 1993.

[8] J. Tippayachai, W. Ongsakul, and I. Ngamroo, "Parallel micro genetic algorithm for constrained economic dispatch," *IEEE Trans. Power Syst.*, vol. 17, no. 3, pp. 790-797, Aug 2003.

[9] N. Sinha, R. Chakrabarti, and P. K. Chattopadhyay, "Evolutionary programming techniques for economic load dispatch," *IEEE Evol. Comput.*, vol. 7, no. 1, pp. 83-94, Feb 2003.

[10] H. T. Yang, P. C. Yang, and C. L. Huang, "Evolutionary programming based economic dispatch for units with nonsmooth fuel cost functions", *IEEE Trans. Power Syst.*, vol. 11, no. 1, pp. 112-118, Feb 1996.

[11] W. M. Lin, F. S. Cheng, and M. T. Tsay, "An improved tabu search for economic dispatch with multiple minima," *IEEE Trans. Power Syst.*, vol. 17, no. 1, pp. 108-112, Feb 2002.

[12] P. Attaviriyanupap, H. Kita, E. Tanaka, and J. Hasegawa, "A hybrid EP and SQP for dynamic economic dispatch with nonsmooth fuel cost function," *IEEE Trans. Power Syst.*, vol. 17, no. 2, pp. 411-416, May 2002.

[13] J. H. Park, Y. S. Kim, I. K. Eom, and K. Y. Lee, "Economic load dispatch for piecewise quadratic cost function using Hopfield neural network," *IEEE Trans. Power Syst.*, vol. 8, no. 3, pp. 1030-1038, Aug 1993.

[14] K. Y. Lee, A. Sode-Yome, and J. H. Park, "Adaptive Hopfield neural network for economic load dispatch," *IEEE Trans. Power Syst.*, vol. 13, no. 2, pp. 519-526, May 1998.

[15] 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, "Closure to discussion of 'Particle swarm optimization to solving the economic dispatch considering the generator constraints'," *IEEE Trans. Power Syst.*, vol. 19, no. 4, pp. 2122-2123, Nov 2004.

[16] D. N. Jeyakumar, T. Jayabarathi, and T. Raghunathan, "Particle swarm optimization for various types of economic dispatch problems," *Electrical Power and Energy Systems*, vol. 28, pp. 36-42, 2006.

[17] T. O. Ting, M. V. C. Rao, and C. K. Loo, "A novel approach for unit commitment problem via an effective hybrid particle swarm optimization," *IEEE Trans. Power Syst.*, vol. 21, no. 1, pp. 411-418, Feb 2006.

[18] A. I. Selvakumar and K. Thanushkodi, "A new particle swarm optimization solution to nonconvex economic dispatch problems," *IEEE Trans. Power Syst.*, vol. 22, no. 1, pp. 42-51, Feb 2007.

[19] J.-B. Park, K.-S. Lee, J.-R. Shin, and K. Y. Lee, "A particle swarm optimization for economic dispatch with nonsmooth cost functions," *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 34-42, Feb 2005.

[20] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *Proc. IEEE Int. Conf. on Neural Networks*, pp. 1942-1948, Perth, Australia, 1995.

[21] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," *Proc. of the Seventh Annual Conference on Evolutionary Programming*, IEEE Press, 1998.

[22] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," *Proc. of IEEE Conf. on Systems, Man, and Cybernetics*, pp. 4104-4109, 1997

[23] Y. Shi and R. C. Eberhart, "Fuzzy adaptive particle swarm optimization," *IEEE Int. Conf. on Evolutionary Computation*, pp. 101-106, 2001.

[24] D. W. Stephens and J. R. Krebs, "Foraging theory," University press, Princeton, New Jersey, 1986.

[25] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization," University press, Princeton, New Jersey, 2001.

[26] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52-67, Jun 2002.

[27] S. Mishra, "A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation," *IEEE Trans. Evolutionary Computation*, vol. 9, no. 1, pp. 61-73, Feb 2005.

[28] The ILOG CPLEX Website, Available: http://www.ilog.com/products/cplex/.

[29] The Dash XPRESS Website, Available: http://www.dashoptimization.com/home/products/products_optimizer.html.