

Discontinue Economic Load Dispatch problem solving by Bacteria Foraging Optimization

K.B.P.Mahavishnu
Asst. Professor/EEE
E-mail:mahavishnu0011@gmail.com

P.Ragupathy
Asst. Professor/EEE
E-mail: ragupathyp07@gmail.com

Dr M.Belsam Jeba Ananth
Prof/EEE
E-mail:belsam@reddifmail.com

Abstract -- This paper presents an efficient approach for solving economic load dispatch (ELD) problems with discontinue cost functions using bacteria foraging algorithm. The bacterial foraging optimization (BFO) algorithm mimics how bacteria forage over a landscape of nutrients to perform parallel non-gradient optimization [1]. In this article, the author provides a tutorial on BFO, including an overview of the biology of bacterial foraging that models this process. The applications of BFO are presented. The performance of algorithms are investigated on ELD problem of various population size with the power demand constraint, generating limits and the prohibited operating zone also added with the fuel cost function. This leads the non-smooth and complexity having cost curves where conventional gradient-based methods are inapplicable.

Keywords: *Foraging, Swarming, Tumbling, Economic dispatch, Prohibited operating zone.*

I INTRODUCTION

Power system is the one of the complex network in the world. There are many problems are associated with this such as load forecasting, Unit commitment (UC), Economic load dispatch (ELD), etc. the above problems are solved by using various conventional method and Deterministic method,. Where Economic Load Dispatch is an important optimization task in power system operation for allocating generation among the committed units [6] such as the constraints imposed are satisfied and the energy requirements in terms of rupees per hour (Rs/h) are minimized. It (ELDP) is defined as the minimization fuel cost (Objective Function) of generating plants subject to various constraints (Condition). The ELD problem with fuel cost function subject to generating constraint can be solved by using deterministic method such as gradient method, λ -iteration method. The fuel cost function with generating constraint is a linear, convex, smooth and continue function. Improvements in scheduling the unit outputs can lead to significant cost savings. Traditional dispatch algorithms employ Lagrangian multipliers and require monotonically increasing incremental cost curves. Unfortunately, the input-output characteristics of modern units are inherently highly nonlinear because of valve-point loadings, ramp rate limits, prohibited operating zone etc., so the resultant objective function became as highly non-linear, non-convex, and discontinue objective function. And

furthermore they may generate multiple local minimum points in the cost function.

In light of the nonlinear characteristics of the units, there is a demand for techniques that do not have restrictions on the shape of the fuel-cost curves. Classical calculus-based techniques (gradient, λ -iteration) fail to address these types of problems satisfactorily. Unlike some traditional algorithms, dynamic programming (DP) [6] imposes no restrictions on the nature of the cost curves and therefore it can be solved as ELD problems with inherently nonlinear and discontinuous cost curves. This method, however, suffers from the “curse of dimensionality” or local optimality.

So the new way for obtaining solution for these non-linear problems is soft computing methods which include various techniques based on learning from nature. These type of algorithms are formed from the behavior of nature. Some of the techniques are Genetic algorithm (GA), Differential Evolution algorithm (DE), Particle Swarm Optimization (PSO), etc..., Recently Kevin M. Passino introduce a technique [4], which imitate the foraging(searching for food) behavior of E.coli bacterium .

II. ESCHERICHIA - COLI (BIOLOGY)

Structure:

It is commonly found in the lower intestine of warm-blooded organisms (endotherms). Most *E. coli* strains are harmless, but some serotypes can cause serious food poisoning in humans. The *E. coli* (Escherichia Coli) bacterium has a plasma membrane, cell wall, and capsule that contain, for instance, the cytoplasm and nucleoid [4]. The pili (singular, pilus) is used as a type of gene transfer to other *E. coli* bacteria, and flagella (singular, flagellum) which are used for locomotion. The cell is about 1mm in diameter, and 2mm in length. The *E. coli* cell only weighs about 1 Pico-gram, and is composed of about 70% water. Salmonella typhimurium is a similar type of bacterium.

The *E. coli* bacterium has a control system which enables it to search for food and try to avoid noxious substances (the resulting motions are called “taxes”. For instance, it swims away from alkaline and acidic environments, and towards more neutral ones. To explain the motile behavior of *E.coli*

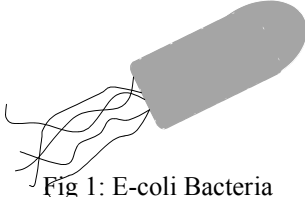


Fig 1: E-coli Bacteria

bacteria, we will explain its actuator (the flagella), “decision-making,” sensors, and closed-loop behavior (i.e., how it moves in various environments its “motile behavior”). You will see that *E. coli* perform a type of “salutatory search.”

2.1 Swimming and tumbling via flagella

Locomotion is achieved via a set of relatively rigid flagella that enable it to “swim” via each of them rotating in the same direction for about 100 200 - revolutions per second. An *E. coli* bacterium can move in two different ways: it can “run” (swim for a period of time) or it can “tumble,” and also it alternates between these two modes of operation[3] for its entire lifetime (i.e., it is rare that the flagella will stop rotating). First, we explain each of these two modes of operation. Following that, we will explain how it decides and also how long to swim before it tumbles.

If the set direction of movement and there is little displacement. To tumble after a run, the cell slows down or stops first; since bacteria are so small they experience almost no inertia, flagella rotate clockwise, each flagellum pulls on the cell and the net effect is that each flagellum operates relatively independent of the others and so the bacterium “tumbles” about (i.e., the bacterium does not have only viscosity, so that when a bacterium stops swimming, it stops within the diameter of a proton.

2.2 Bacterial motile behavior

The motion patterns (called “taxes”) that the bacteria will generate in the presence of chemical attractants and repellents are called chemotaxis.

First, note that if an *E. coli* is in some substance that is neutral in the sense that it does not have food or noxious substances, and if it is in this medium for a long time (e.g., more than 1 min), then the flagella will simultaneously alternate between moving clockwise and counter clockwise so that the bacterium will alternately tumble and run[7]. This alternation between the two modes will move the bacterium, but in random directions, and this enables it to “search” for nutrients.

Next, suppose that the bacterium happens to encounter a nutrient gradient (e.g., serine. The *change* in the concentration of the nutrient triggers a reaction such that the bacterium will spend more time swimming and less time tumbling.

On the other hand, typically if the bacterium happens to swim down a concentration gradient (or into a positive gradient of noxious substances), it will return to its baseline behavior so that essentially it tries to search for a way to climb back up the gradient.

2.3 Underlying sensing and decision-making mechanisms

The sensors are the receptor proteins that are signalled directly by external substances (e.g., in the case for the pictured amino acids) or via the periplasmic substrate-binding proteins. The receptor proteins then affect signalling molecules inside the bacterium[3]. Also, there is an effect of “adding machine” and an ability to compare values to arrive at an overall decision about which mode the flagella should operate in; essentially, the different sensors add and subtract their effects, and the more active or numerous will have a greater influence on the final decision.

2.4 Elimination and dispersal events

It is possible that the local environment where a population of bacteria live changes either gradually (e.g., via consumption of nutrients) or suddenly due to some other influence. Events can occur such that all the bacteria in a region are killed or a group is dispersed into a new part of the environment. What is the effect of elimination and dispersal events on chemotaxis? They have the effect of possibly destroying chemotactic progress, but they also have the effect of assisting in chemotaxis, since dispersal may place bacteria near good food sources. From a broad perspective, elimination and dispersal are parts of the population-level long-distance motile behavior.

2.5 *E. coli* bacterial swarm foraging for optimization

Suppose that we want to find the minimum of $J(\theta)$, $\theta \in \mathbb{R}^p$, where we do not have measurements or an analytical description of the $J(\theta)$ [5]. Here, we use ideas from bacterial foraging to solve this non gradient optimization problem. First, suppose that θ is the position of a bacterium and $J(\theta)$ represents the combined effects of attractants and Repellents from the environment, with, for example, $J(\theta) < 0$, $J(\theta) = 0$, and $J(\theta) > 0$ representing that the bacterium at location θ is in nutrient-rich, neutral, and noxious environments[8], respectively. Basically, chemotaxis is a foraging behavior which implements a type of optimization where bacteria try to climb up the nutrient concentration (find lower and lower values of $J(\theta)$), avoid noxious substances, and search for ways out of neutral media (avoid being at positions θ where $J(\theta) \geq 0$). It implements a type of biased random walk.

III. PROBLEM FORMULATION

The ELD load dispatch problem can be described as an optimization (minimization) process with the objective:

$$\text{Min} \sum_{j=1}^N F_j(P_j) \quad (1)$$

Where $F_j(P_j)$ is the fuel cost of the thermal unit ‘J’, which is the function of P_j . normally fuel cost function of a thermal unit is quadratic, sometimes it may consider as the cubic function.

$$F_j(P_j) = a_j \times P_j^2 + b_j \times P_j + c_j \quad (2)$$

And the above objective function is subjected to these constraints.

Power balance: This is a equality constraint for this objective function

$$\sum_{j=1}^n P_j - (P_D + P_L) = 0 \quad (3)$$

Where P_D is the system load demand and P_L is the transmission Loss

$$P_L = \sum_{i=1}^N \sum_{j=1}^N P(i) \times B(i, j) \times P(j) + \sum_{i=1}^N P(i) \times B_0(i) + B_{00} \quad (4)$$

Generating Capacity Constraints:

This is a generating limit constraint

$$P_{JMIN} \leq P_J \leq P_{JMAX} \text{ For } J=1,2,\dots,n \quad (5)$$

Where P_{JMIN} and P_{JMAX} are the minimum and maximum power outputs of the unit.

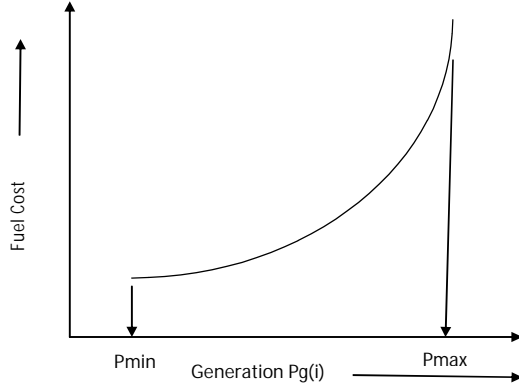


Fig 2: Normal Fuel Cost Function

Valve point effect

The fuel-cost function considering valve-point loadings [6] of the generating units are given as

$$FC_J(P_J) = a_J \times P_J^2 + b_J \times P_J + c_J + e_J \times \sin(f_J(P_{JMIN} - P_J)). \quad (5)$$

Where a_J , b_J , and c_J are the fuel-cost coefficients of the unit, and e_J and f_J are the fuel cost coefficients of the unit with valve-point effects. The generating units with multivalve steam turbines exhibit a greater variation in the fuel-cost functions. The valve-point effects introduce ripples in the heat-rate curves.

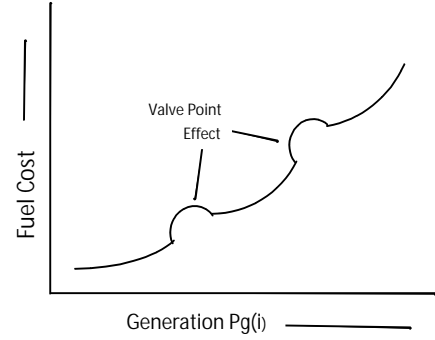


Fig 3: Fuel Cost Function with Valve Point Effect

Prohibited operating zone

Normally a generating unit is designed for running in pre-specified range between minimum and maximum generating limits for protective operation. In this region the unit seems to be a continuous function.

But in some special cases, the unit seems that it undergoes for a mechanical vibration when it is operating in a particular short region. Mechanical vibrations cause cumulative metal fatigue in turbine blades and lead to premature turbine blade failures. Because of these problems the unit should not operating on those region, is called as Prohibited operating zone.

Due to this the continuous fuel cost function is become as a discontinuous cost function that leads to complexity of the problem.

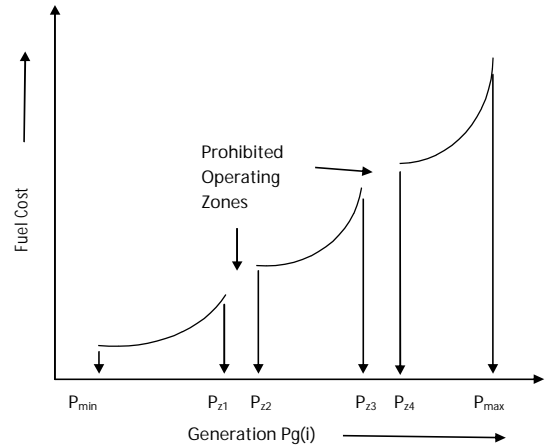


Fig 4: Fuel Cost Function with Prohibited Operating Zone

IV. STEPS INVOLVED

The foraging strategy of E-coli bacteria is converted as a mathematical process for solution finding of various problems. It is governed basically by four processes namely Chemotaxis, Swarming, Reproduction, Elimination and Dispersal.

4.1. Chemotaxis

Define a chemotactic step to be tumble followed by a tumble or a tumble followed by a run. Let j be the index for the chemotactic step. Let k be the index for the reproduction step. Let l be the index of the elimination-dispersal event.

$$P(j, k, l) = \{\theta^i(j, k, l) | i = 1, 2, \dots, S\} \quad (6)$$

let

$J(i, j, k, l)$ denote the cost at the location of the i^{th} bacterium $\theta^i(j, k, l) \in \mathbb{R}^p$

It represents the position of each member in the population of the S bacteria at the j^{th} chemotactic step, k^{th} reproduction step, and l^{th} elimination-dispersal event.

To represent a tumble, a unit length random direction, say $\phi(j)$, is generated; this will be used to define the direction of movement after a tumble. In particular, we let

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \phi(j) \quad (7)$$

Where

$$\phi(j) = \frac{\Delta(i)}{\sqrt{\Delta^T(i) \times \Delta(i)}} \quad (8)$$

$C(i)$ = step size for random direction

If at $\theta^i(j+1, k, l)$ the cost $J(i, j+1, k, l)$ is better (lower) than at $\theta^i(j, k, l)$, then another step of size $C(i)$ in this same direction will be taken, and again, if that step resulted in a position with a better cost value than at the previous step, another step is taken. This swim is continued as long as it continues by the (3.2) to reduce the cost, but only up to a maximum number of steps, $N_s(4)$.

4.2. Swarming

Bacteria exhibits swarm behavior i.e. healthy bacteria try to attract other bacteria so that together they reach the desired location (solution point) more rapidly. The effect of Swarming is to make the bacteria congregate into groups and move as concentric patterns with high bacterial density. Mathematically swarming behavior can be modeled as (7)

It also have cell – to – cell signaling via an attractant and will represent that with $J_{cc}^i(\theta, \theta^i(j, k, l))$, $i = 1, 2, \dots, S$, for the i^{th} bacterium This produces the swarming effect. When we want to study swarming, the i^{th} bacterium, $i = 1, 2, \dots, S$, will hill-climb on

TABLE 1 PARAMETERS FOR SWARMING

S.No	Parameter	Value
1	d_{attract}	0.1
2	w_{attract}	0.2
3	$h_{\text{repellent}}$	0.2
4	$w_{\text{repellent}}$	10

$$J_{cc}(\theta, P(j, k, l)) = \sum_{i=1}^S J_{cc}^i(\theta, \theta^i(j, k, l))$$

$$= \sum_{i=1}^S \left[-d_{\text{attract}} \exp \left(-w_{\text{attract}} \sum_{m=1}^p (\theta_m - \theta_m^i)^2 \right) \right]$$

$$+ \sum_{i=1}^S \left[-h_{\text{repellent}} \exp \left(-w_{\text{repellent}} \sum_{m=1}^p (\theta_m - \theta_m^i)^2 \right) \right]$$

9)

$$J = J(i, j, k, l) + J_{cc}(\theta, P) \quad (10)$$

So that the cells will try to find nutrients, avoid noxious substances, and at the same time try to move toward other cells, but not too close to them. The $J_{cc}(\theta)$ function dynamically deforms the search landscape as the cells move to represent the desire to swarm (i.e., we model mechanisms of swarming as a minimization process).

4.3 Reproductions

After N_c chemotactic steps, a reproduction step is taken. Let N_{re} be the number of reproduction steps to be taken. For convenience, we assume that S is a positive even integer.

$$S_R = S/2 \quad (11)$$

Let be the number of population members who have sufficient nutrients so that they will reproduce (split in two) with no mutations. For reproduction, the population is sorted in order of ascending accumulated cost then the S_R least healthy bacteria die and the other S_r healthiest bacteria each split into two bacteria, which are placed at the same location.

4.4. Elimination and Dispersal

Let N_{ed} be the number of elimination-dispersal events, and for each elimination-dispersal event each bacterium in the population is subjected to elimination-dispersal (1). We assume that the frequency of chemotactic steps is greater than the frequency of reproduction steps, which is in turn greater in frequency than elimination-dispersal events.

V. ALGORITHM

Step 1: Initializations

For initialization, choose p , S , N_c , N_s , N_{re} , N_{ed} , p_{ed} , and the $C(i)$, $i = 1, 2, \dots, S$.

Choose initial values for the θ_i , $i = 1, 2, \dots, S$,

Step 2: Elimination-dispersal loop: $l = l + 1$

Step 3: Reproduction loop: $k = k + 1$

Step 4: Chemotaxis loop: $j = j + 1$

Step 4.1: compute the fitness value of i^{th} bacteria

$$J(i, j) = J(i, j) + J_{cc} \text{ take } J(i, j) = J_{\text{last}}$$

The value of J_{cc} found by using eqn (8).

Step 4.2: Tumble

find new position of i^{th} bacteria by eqn(7) and (8)

Find the fitness value $J(i,j+1)$ and compare,

If $J(i,j+1) < J_{last}$ Set

$J_{last} = J(i,j+1)$ else $J_{last} = J_{last}$

Step 4.3: Swimming

set $m=0$ let $m=m+1$ find new position by eqn (8)

Find the new position fitness value $J(i,j+1)$ and

if $J(i,j+1) < J_{last}^i$ set $J_{last}^i = J(i,j+1)$ Else set $J_{last}^i = J_{last}^i$

Step 4.4: termination criteria

Check if $m < N_s$ then go to step 4.3 else set $i=i+1$

Check if $i < S$ then go to step 4.1 else

Check if $j < N_c$ then go to step 4. otherwise go to step 5 .

Step 5: Reproduction:

Find the health of the each bacteria Sort the each bacteria by its cost.

Lowest cost \rightarrow highest health (split into two)

Highest cost \rightarrow lowest health (will die)

check if $k < N_{re}$ go to step 3 else go to step 6.

Step 6: Elimination/dispersal

Eliminate the lowest health bacteria after reproduction and disperse it to a new random position. Check if $l < N_{ed}$ go to step 2 else terminate process.

6. RESULT ANALYSIS

The results were analyzed for a case study of system with six units. The objective function added with losses and prohibited operating zones. The fuel cost coefficient a , b and c are given in below table.

TABLE 2: FUEL COST COEFFICIENT

UNIT	P_{MIN}	P_{MAX}	A_i	B_i	C_i
1	100	500	0.0070	7.0	240
2	50	200	0.0095	10.0	200
3	80	300	0.0090	8.5	220
4	50	150	0.0090	11.0	200
5	50	200	0.0080	10.5	220
6	50	120	0.0075	12.0	190

The ramp rate limits and prohibited operating zones are values with the initial generations are given in below table.

TABLE 3: PROHIBITED OPERATING ZONE LIMITS

UNIT	P_i^0	UR	DR	PROHIBITED ZONE	
				ZONE 1	ZONE 2
1	440	80	120	[210 240]	[350 380]
2	170	50	90	[90 110]	[140 160]
3	200	65	100	[150 170]	[210 240]
4	150	50	90	[80 90]	[110 120]
5	190	50	90	[90 110]	[140 150]
6	110	50	90	[75 85]	[100 105]

And the coefficient of the power loss equation B, B_0, B_{00}

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

$$B_0 = 1.0 \times 10^{-3} [-0.3908 -0.1297 0.7047 0.0591 0.2161 -0.6635]$$

$$B_{00} = 0.056$$

The results analyzed for the various size of population with different iteration value are compared.

TABLE 4: RESULT ANALYSIS FOR VARIOUS CASE

ITERATION	20	40	50	100
POPULATION				
20	15466.56	15464.10	15500.33	15466.16
50	15542.36	15472.18	15475.75	15468.37
100	15464.38	1541.54	15451.23	15460.28

From the table we decide that the solution accuracy doesn't affected by improving the iteration, only by the population size

The result for the study case for the six generating units, total fuel cost including the power loss are tabulated as 50 trials of solution for 100 population and 50 iteration

TABLE 5: GENERATION SOLUTION FOR UNITS

UNITS	VALUE
P_1	439.9741
P_2	185.9209
P_3	250.6985
P_4	124.58
P_5	189.5803
P_6	85.20
P_{DEMAND}	1275.969MW
P_{LOSS}	12.969MW
TOTAL FUEL COST	Rs.15454.26

7. CONCLUSION

The foraging behavior of the bacteria solves the power system problems with more local optimum. This is fully a stochastic (random) based algorithm. The algorithm consumes

more time to find solution due to the more inner loops circulated inside algorithm. The result of the problem does not depend on the iteration but on population. Comparing with other soft computing technique BFO finds global optimum, but the precision of the solution is very low, even though for higher iteration. So it concluded as the algorithm can be modified to improve the speed of algorithm by reducing the cycles.

REFERENCE

- 1) Kevin M. Passino "Bio Mimicry Of Bacterial Foraging "
- 2) Dong Hwa Kim A, Ajith Abraham B, And , Jae Hoon Cho A "A Hybrid Genetic Algorithm And Bacterial Foraging Approach For Global Optimization" On *Information Sciences* 177 (2007) 3918–3937 Received 14 June 2005; Received In Revised Form 22 March 2007; Accepted 2 April 2007
- 3) J. Hazra, Student Member, *IEEE*, And A. K. Sinha, Member, *IEEE* "Environmental Constrained Economic Dispatch Using Bacterial Foraging Technique".
- 4) Kaveh Aboukazar, And Mohd Wazir Mustafa, Member, " Unit Commitment Optimization Using Improved Genetic Algorithm" *IEEE Paper Accepted For Presentation At 2009 IEEE Bucharest Power Tech Conference*, June 28th - July 2nd, Bucharest, Romania
- 5) Kevin M. Passino, The Ohio State University, USA "Bacterial Foraging Optimization" *International Journal Of Swarm Intelligence Research*, 1(1), 1-16, January-March 2010 Copyright © 2010, IGI.
- 6) Nidul Sinha, R. Chakrabarti, And P. K. Chattopadhyay "Evolutionary Programming Techniques For Economic Load Dispatch " *IEEE Transactions On Evolutionary Computation*, Vol. 7, No. 1, February 2003
- 7) Sibylle D. Müller, Jarnomarchetto, Stefano Airaghi, And Petros Koumoutsakos "Optimization Based On Bacterial Chemotaxis" On *IEEE TRANSACTIONS ON Evolutionary Computation*, Vol. 6, No. 1, February 2002
- 8) Swagatam Das, Sambartadasgupta, Arijitbiswas, Ajith Abraham, Senior Member, *IEEE*, And Amitkonar, Member, *IEEE* "On Stability Of The Chemotactic Dynamics In Bacterial-Foraging Optimization Algorithm" On *IEEE Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans*, Vol. 39, No. 3, May 2009
- 9) Weibing Liu , Min Li, Xianjia Wang "An Improved Particle Swarm Optimization Algorithm For Optimal Power Flow"
- 10) Y. LIU, K. M. Passino And M. A. Simaan "Biomimicry Of Social Foraging Bacteria For Distributed Optimization: Models, Principles, And Emergent Behaviours" On *Journal Of Optimization Theory And Applications*: Vol. 115, No. 3, Pp. 603–628, December 2002 (2002).

Acknowledgement:

We (Authors) are heartily extend our thanks to our friend as well as mentor **Mr. Sivakumar**, who guide us, teach us for our future and knowledge with great of wish.



Mahavishnu obtained his B.E., in Electrical-Electronics at pallavan college of engineering (2009) , M.E., in Power system engineering at Anna University of technology Coimbatore (2012) , Tamilnadu, India. Now he is currently working as Asst. Professor in engineering college near Chennai. His areas of interest are in the field of Power system operation, soft computing and optimization.



Ragupathy obtained his B.E., in Electrical-Electronics at Govt. Engineering college, M.E., in Power system engineering at Anna University of technology Coimbatore (2012), Tamilnadu, India. Now he is currently working as Asst. Professor in engineering college near Chennai. His area of interest is in the field of Economic load dispatch and soft computing.