

# Optimal Power Flow Using Dynamic Bacterial Forging Algorithm

Hema Sharma, Ilyas, Suryakant

**Abstract**— Optimal power flow (OPF) problem has already been attempted as a static optimization problem, by adopting conventional gradient-based methods and more recently, no conventional ones, such as evolutionary algorithms. However, as the loads, generation capacities and network connections in a power system are always in a changing status, these static-oriented methods are of limited use for this issue. This paper presents a new algorithm, dynamic bacterial foraging algorithm (DBFA), for solving an OPF problem in a dynamic environment in which system loads are changing. DBFA is based on the recently proposed BFA which mimics the basic foraging behaviour of *E. coli* bacteria. A selection scheme for bacteria's reproduction is employed in DBFA, which explores the self-adaptability of each bacterium in the group searching activities. DBFA has been evaluated, for optimizing the power system fuel cost with the OPF embedded, on the standard IEEE 30-bus with a range of load changes which occurred in different probabilities. The simulation results show that DBFA can more rapidly adapt to load changes, and more closely trace the global optimum of the system fuel cost, in comparison with BFA and some other techniques.

**Index Terms**—Bacterial foraging algorithm (BFA), Optimal Power Flow, Dynamic Bacterial foraging algorithm

## I. INTRODUCTION

The purpose of OPF is to find the optimal settings of a given power system network that optimize a certain objective function while satisfying its power flow equations, system security, and equipment operating limits. Different control variables are manipulated to achieve an optimal network setting based on the problem formulation. To tackle complex search problems of the real world, scientists have been drawing inspiration from nature and natural creatures for years. Optimization techniques may follow different approaches. Recently, optimization techniques inspired by biology behaviours, known as bio-mimetic optimization algorithms, have obtained more and more attention. Some algorithms have been proposed, such as Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and glow-worm swarm optimization (GSO). Bio-mimetic optimization algorithms are developed from simulation the evolutionary process and the behaviours of biology. They are population-based (each member stands for a biology individual), and initialized with a population of individuals.

Manuscript published on 30 June 2013.

\*Correspondence Author(s)

Hema Sharma, Department Of Electrical & Electronics, Al-Falh School of Engineering & Technology, India.

Mohd. Ilyas, Assistant Professor, Deptt. Of Electrical & Electronics, Al-Falh School of Engineering & Technology, India.

Suryakant, Assistant Professor, Department Of Electrical & Electronics, HMRITM, Delhi, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

They utilize the direct information "fitness" instead of individual's ability to adapt to the environment. These individuals are manipulated over many generations by ways of mimicking social behaviour of biology, in an effort to find the optima. Natural selection tends to eliminate animals with poor foraging strategies and favour the propagation of genes of those animals that have successful foraging strategies, since they are more likely to enjoy reproductive success. After many generations, poor foraging strategies are either eliminated or shaped into good ones. This activity of foraging led the researchers to use it as optimization process. Natural selection tends to eliminate species with poor foraging strategies and favour the propagation of genes of species with successful foraging behaviour since they are more likely to enjoy reproductive success. Since a foraging organism or animal takes necessary action to maximize the energy intake per unit time spent for foraging, considering all the constraints presented by its own physiology such as sensing and cognitive capabilities, environment (e.g., density of prey, risks from predators, physical characteristics of the search space), the natural foraging strategy can lead to optimization and essentially this idea can be applied to solve real-world optimization problems.

In 2002 K. M. Passon proposed Bacterial Foraging optimization Algorithm which is inspired by the Foraging behaviour of *E. coli* bacteria. Natural selection tends to eliminate animals with poor foraging strategies and favour the propagation of genes of those animals has successful foraging strategies. Bacteria foraging strategies are the motivation for BFA optimization.

**Foraging Theory:** - Animals search for and obtain nutrients in a way that maximizes the ratio  $E/T$  (where  $E$  is the energy obtained and  $T$  is the time spent foraging) or maximizes the long-term average rate of energy intake. Evolution optimizes the foraging strategies, since animals that have poor foraging performance do not survive.

In the process of foraging, *E. coli* bacteria undergo four stages, namely, chemotaxis, swarming, reproduction, and elimination and dispersal. In search space, BFOA seek optimum value through the chemotaxis of bacteria, and realize quorum sensing via assemble function between bacterial, and satisfy the evolution rule of the survival of the fittest make use of reproduction operation, and use elimination-dispersal mechanism to avoiding falling into premature convergence.

### A. Chemotaxis-

This process simulates the movement of an *E. coli* cell through swimming and tumbling via flagella. Biologically, an *E. coli* bacterium can move in two different ways.



It can swim for a period of time in the same direction, or it may tumble, and alternate between these two modes of operation for the entire lifetime. Suppose  $\theta(i, j, k, l)$  represents  $i$  th bacterium at  $j$  the chemotactic,  $k$ th reproductive and  $l$  th elimination dispersal step.  $C(i)$  is the size of the step taken in the random direction specified by the tumble (run length unit). Then in computational chemotaxis the movement of the bacterium maybe represented by

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

where  $\Delta$  indicates a vector in the random direction whose elements lie in  $[-1, 1]$ .

### B. Swarming-

interesting group behaviour has been observed for several motile species of bacteria including *E.coli* and *S. typhimurium*, where intricate and stable spatiotemporal patterns (swarms) are formed in a semisolid nutrient medium. A group of *E.coli* cells arrange themselves in a travelling ring by moving up the nutrient gradient when placed amidst a semisolid matrix with a single nutrient chemo effector. The cells, when stimulated by a high level of *succinate*, release an attractant *aspartate*, which helps them to aggregate into groups and thus move as concentric patterns of swarms with high bacterial density. The cell-to-cell signalling in *E. coli* swarm may be represented by the following function:

$$J_{cc}(\theta, P(j, k, l)) = \sum_{i=1}^S J_{cc}^i(\theta, \theta^i(j, k, l)) \\ = \sum_{i=1}^S [-d_{attract} \exp(-w_{attract} \sum_{m=1}^P (\theta_m - \theta_m^i)^2)] \\ + \sum_{i=1}^S [h_{repellant} \exp(-w_{repellant} \sum_{m=1}^P (\theta_m - \theta_m^i)^2)]$$

where the cost function value is added to the actual cost function.  $S$  is the total number of bacteria and  $p$  is the number of parameters to be optimized which are present in each bacterium.  $d_{attract}$  is the depth of the attractant released by the cell and  $w_{attract}$  is a measure of the width of the attractant signal.  $h_{repellant} = d_{attract}$  is the height of the repellent effect and  $w_{repellant}$  is a measure of the width of the repellent.

### C. Reproduction-

According to the rules of evolution, individual will reproduce themselves in appropriate conditions in a certain way. For bacterial, a reproduction step takes place after all chemotactic steps.

$$J_{health}^i = \sum_{j=1}^{N_c+1} J(i, j, k, l)$$

Where  $J_{health}^i$  is the health of bacterium  $i$ . Sort bacteria and chemotactic parameters  $C(i)$  in order of ascending cost (higher cost means lower health). For keep a constant population size, bacteria with the highest  $J_{health}$  values die.

The remaining bacteria are allowed to split into two bacteria in the same place.

### D. Elimination-Dispersal-

In the evolutionary process, elimination and dispersal events can occur such that bacteria in a region are killed or a group is dispersed into a new part of the environment due to some influence. 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 the evolutionary point of view, elimination and dispersal was used to guarantees diversity of individuals and to strengthen the ability of global optimization. In BFA, bacteria are eliminated with a probability of  $P_{ed}$ . In order to keeping the number of bacteria in the population constant, if a bacterium is eliminated, simply disperse one to a random location on the optimization domain.

## II. DYNAMIC BACTERIAL FORAGING ALGORITHM

The BFA is inspired by the pattern exhibited by bacterial foraging behaviour. Bacteria have the tendency to gather to the nutrient-rich areas by an activity called “chemotaxis”. It is known that bacteria swim by rotating whip-like flagella driven by a reversible motor embedded in the cell wall. One example is *E. coli*, which has 8–10 flagella placed randomly on its body. When all flagella rotate counter clockwise, they form a compact, helically propelling the cell along a helical trajectory, which is called Run. When the flagella rotate clockwise, they all pull on the bacterium in different directions, which causes the bacteria to Tumble. Apart from the bacterial chemotaxis, BFA also introduces reproduction as well as elimination and dispersion events. However, in DBFA, there are only two primary components: local search and selection process.

**Local Search-** Bacteria make decision according to their ambient environment. The motion of individual peritrichous flagellated bacterium can be described in terms of run intervals during which the cell swims approximately in a straight line interspersed with tumbles, when the organism undergoes a random reorientation.

After one step move, the position of the  $i$ th bacterium can be represented as

$$\theta(i, j+1, k, l) = \theta(i, j, k, l) + C(i, j, k, l) \angle \phi(i, j, k, l)$$

Where  $\theta(i, j, k, l)$  indicates the position of the  $i$ th bacterium at the  $j$ th chemotactic step in the  $k$ th reproductive loop of the  $l$ th elimination and dispersion event;  $C(i, j, k, l)$  is the length of a unit walk, which is set to be a constant; and  $\phi(i, j, k, l)$  is moving angle of the  $i$ th bacterium. When its activity is Run,  $\phi(i, j, k, l)$  is the

same with  $\phi(i, j-1, k, l)$  otherwise,  $\sum_{i=1}^P W_i = 1$  is random

angle ranged from  $[0, 2\pi]$ . In this case, the fitness evaluation is calculated as  $J(i, j, k, l)$ .



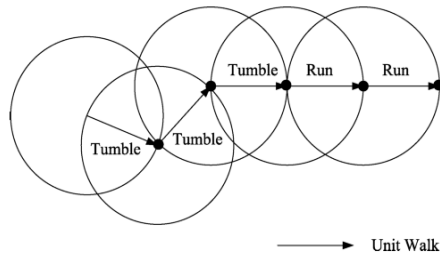


Fig.1. Tumble and Run

**Selection Process-** Selection process is introduced using a more flexible scheme to enable a better adaptability in a changing environment. The scheme is described as follows:

$$F_i = \sum_{j=1}^{N_c} J(i, j, k)$$

$$R_i = \text{sort}(F_i)$$

$$W_i = m \frac{(R_i)^n}{\sum_{i=1}^P (R_i)^n} + (1-m) \frac{F_i}{\sum_{i=1}^P F_i}$$

Where  $N_c$  is the number of chemotactic steps (each step may contain one Run or Tumble) during a bacterium's life time,  $j$  the index of chemotactic steps,  $P$  the population size,  $m$  the weight of diversity, and  $n$  the exponent of  $R_i$ . Here the nutrient obtained by bacterium  $i$  at step  $j$  is denoted as  $J(i, j, k)$  and the sum of nutrient obtained during life is indicated in (3.5). Thus, those experiencing more nutrient-rich areas are more likely to be selected as a parent for next generation.

#### Step by step procedure of DBFA-OPF

**Step.1** Prepare the database for the generator data, bus data, transformer data and transmission line data.

#### Step.2 Initialization

- Number of parameters ( $p$ ) to be optimized.
- Number of bacteria ( $S$ ) to be used for searching the total region.
- Swimming length  $N_s$  after which tumbling of bacteria will be undertaken in a chemotactic loop.
- $N_c$  the number of iterations to be undertaken in a chemotactic loop. ( $N_c > N_s$ ).
- $N_{re}$ , the maximum number of reproduction to be undertaken.
- The location of each bacterium  $P(1-p, 1-s, 1)$  which is specified by random numbers on  $[-1, 1]$ .
- The value of  $C(i)$  which is assumed to be constant.
- The values of  $d_{attract}$ ,  $w_{attract}$ ,  $h_{repellant}$  and  $w_{repellant}$

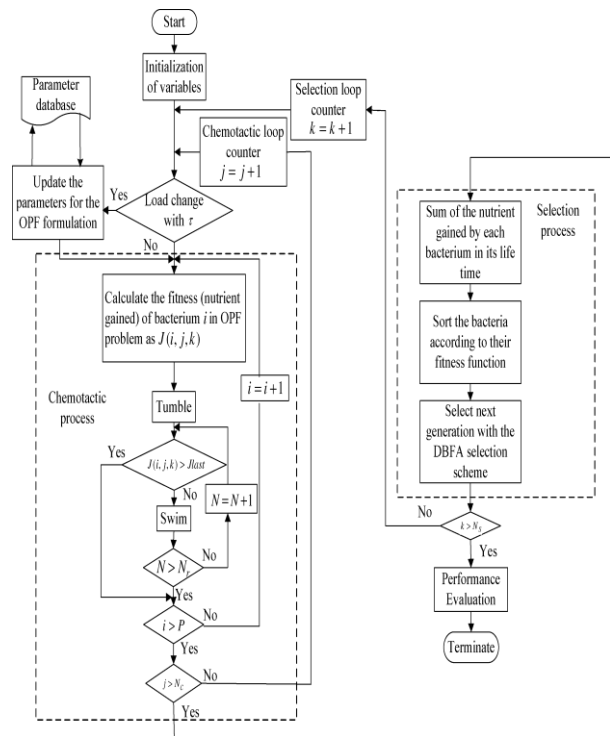


Fig.2. Flow chart of DBFA

#### Step.3 Initialization of loops

- Chemotactic loop counter,  $j=1$
- Reproduction loop counter,  $k=1$

**Step.4** Run power flow using the Newton–Raphson method for each set of generating patterns  $P_{gi}$  corresponding to a particular generation and hence determine, slack bus generation, bus voltage magnitudes and phase angles at all the buses. Also calculate power flow in each transmission line of the system.

#### Step.5 Check the following constraints:

- Check the  $P_{g_{slack}}$ , it should be within  $P_{g_{slack}}^{\min}$  and  $P_{g_{slack}}^{\max}$ .
- Check the bus voltage violation  $V_i^{\min} \leq V_i \leq V_i^{\max}$
- Check the bus voltage phase angle  $\phi_i^{\min} \leq \phi_i \leq \phi_i^{\max}$
- Check the MVA flows violation,  $MVA_{ij} \leq MVA_{ij}^{\max}$  for all the lines connected between bus  $j$  and  $i$ .
- Check real and reactive power limits at all generator buses. If any of the above constraints is violated, go to step 3.

**Step.6** if all the constraints are satisfied, than start iterative procedure in step 7.

#### Step.7 Iterative Procedure

- For  $i=1, 2, \dots, S$ , calculate cost function value for each bacterium  $i$  as follows.

- Compute value of cost function  $J(i, j, k, l)$ . Let  $J_{sw}(i, j, k, l) = J(i, j, k, l) + J_{cc}(\theta^i(j, k, l), P(j, k, l))$  (i.e., add on cell-to-cell attractant effect for swarming behavior).
  - Let  $J_{last} = J_{sw}(i, j, k, l)$  to save this value since we may find a better cost via a run.
  - End for stop loop.
  - For  $i=1, 2, \dots, S$  take the tumbling/swimming decision
  - Tumble: Generate a random vector  $\Delta(i) \in \mathbb{R}^p$  with each element  $\Delta_m(i)$   $m=1, 2, \dots, p$ , a random number on  $[-1, 1]$ .
  - Move:
$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i) \Delta(i)}}$$
- Let
- Compute  $J(i, j+1, k, l)$  and then let  $J_{sw}(i, j+1, k, l) = J(i, j+1, k, l) + J_{cc}(\theta^i(j+1, k, l), P(j+1, k, l))$
  - Swim:
    - Let  $m=0$ ; (counter for swim length)
    - While  $m < N_s$  (have not climbed down too long)
  - Let  $m=m+1$
  - If  $J_{sw}(i, j+1, k, l) < J_{last}$  (if doing better), let  $J_{last} = J_{sw}(i, j+1, k, l)$  and let
$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i) \Delta(i)}}$$
and use this  $\theta^i(j+1, k, l)$  to compute the new  $J(i, j+1, k, l)$ 
    - Else, let  $m=N_s$ . This is the end of the while statement.
    - Go to the next bacterium  $(i+1)$  if  $i \neq S$  to process the next bacterium.
    - If  $j < N_c$  then go to step 3 else move to step 8.

Step.8 Calculate the fitness (nutrient gradient) of bacterium  $i$ , as follows:

$$F_i = \sum_{j=1}^{N_c} J(i, j, k) \quad (1)$$

$$R_i = \text{sort}(F_i) \quad (2)$$

$$W_i = m \frac{(R_i)^n}{\sum_{i=1}^P (R_i)^n} + (1-m) \frac{F_i}{\sum_{i=1}^P F_i} \quad (3)$$

Where  $j$  is the index of chemotactic steps,  $P$  the population size,  $m$  the weight diversity, and  $n$  the exponent of  $R_i$ . Here, the nutrient obtained by bacterium  $i$  at step  $j$  is denoted as  $J(i, j, k)$  and the sum of nutrient obtained during its life is indicated in equation (1). Thus, those experiencing more nutrient-rich areas are more likely to be selected as a parent for next generation. Next, the whole population is sorted according to  $F_i$  using an operator  $\text{sort}$ , and then  $R_i$  is allocated as the rank of bacterium  $i$  in equation (2). Here  $m$  is used to maintain diversity and speed up convergence in the evolution process at the same time. This parameter combines the rank of the bacterium  $(R_i)^n$  with the fitness calculation  $F_i$ , in order to obtain a balance of these two

elements. Consequently, the survival probability of bacterium  $i$ ,  $W_i$  is given in, and  $\sum_{i=1}^P W_i = 1$ .

Step.9 Roulette wheel selection is adopted to generate the next generation, as follows:

- Calculate the total fitness value

$$F_t = \sum_{i=1}^S F_i$$

- Calculate the probability of selection using the formula

$$p_i = \frac{F_i}{F_t} \text{ for } i = \text{number of bacteria, } S$$

- Determine cumulative probabilities for each chromosome

$$q_i = \sum_{j=1}^i p_j \text{ for } i = \text{number of bacteria, } S$$

- Generate random number  $r_i$  (for  $i = 1, S$ ) in the range  $\{0, 1\}$ .
- if  $r_i < q_i$  then select the first chromosome; otherwise select the  $m$ th chromosome such that  $q_{m-1} < r_i < q_m$ .

Step.10 Run power flow using the Newton–Raphson method for each set of generating patterns  $P_{gi}$  corresponding to a particular generation and hence determine, slack bus generation, bus voltage magnitudes and phase angles at all the buses. Also calculate power flow in each transmission line of the system and check for the constraints in step 5.

Step.11 If  $k > N_s$  the go for the performance evaluation and print results else go to step 3.

### III. SIMULATION RESULTS

The DBFA is applied on IEEE-30 Bus system consisting of 6 Generators, 21 Load Buses, 41 Branches, 2 Shunts and 4 Transformers. DBFA is tested and Results are compared with GA & EP on the standard IEEE-30 bus system. The OPF results using DBFA are evaluated using MATLAB (R2009b).

The simulation results obtained using DBFA technique is shown in Table 1. The table also shows the Power Generation of 6 generating Units and total generation cost for IEEE-30 Bus System. Further the results obtained from DBFA is compared with other computational methods, compared results are shown in Table 2.

Table 1: Simulation results of DBFA-OPF

Generator	Bus Number	Power Generation(MW)
G1	1(Reference)	175.8334
G2	2	48.8845
G3	5	24.7843
G4	8	20.2196
G5	11	12.1283
G6	13	12.3373
Total Generation Cost		802.447297 \$/hr

Table 2: Comparison with Other Techniques

S. No	Approach	Minimum Generation Cost
1	GA-OPF [34]	803.915839 \$/hr
2	MATLAB optimization Tool Box [34]	803.550000 \$/hr
3	EP-OPF [34]	803.571862 \$/hr
4	DBFA-OPF	802.447297 \$/hr

#### IV. CONCLUSIONS

The effectiveness of DBFA is verified through simulation results using IEEE-30 bus system which shows that our proposed method is better than GA, EP and MATLAB optimization toolbox.

#### REFERENCES

1. K. M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Syst. Mag., vol. 22, no. 3, pp. 52–67, Jun. 2002.
2. Y. Liu and K. M. Passino, Biomimicry of social foraging bacteria for distributed optimization: Models, principles, and emergent behaviours, J. Optimization Theory Applicat., vol. 115, no. 3, pp. 603–628, Dec. 2002.
3. D. H. Kim, A. Abraham, and J. H. Cho, A hybrid genetic algorithm and bacterial foraging approach for global optimization, Inform. Sci., vol. 177, no. 18, pp. 3918–3937, 2007.
4. S. Mishra, A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation, IEEE Trans. Evol. Comput., vol. 9, no. 1, pp. 61–73, Feb. 2005.
5. M. Tripathy, S. Mishra, L. L. Lai, and Q. P. Zhang, Transmission loss reduction based on FACTS and bacteria foraging algorithm, in Proc. PPSN, 2006, pp. 222–231.
6. S. Mishra, and C. N. Bhende, Bacterial foraging technique-based optimized active power lter for load compensation, IEEE Trans. Power Delivery, vol. 22, no. 1, pp. 457–465, Jan. 2007.
7. D. H. Kim and C. H. Cho, Bacterial foraging based neural network fuzzy learning, in Proc. IICAI 2005, pp. 2030–2036.
8. B. D. Hughes, Random Walks and Random Environments: Volume 1: Random Walks. London, U.K.: Oxford Univ. Press, 1996.
9. M. Avriel, Nonlinear Programming: Analysis and Methods. 1sted. New York: Dover, 2003. ch. 10, sec. 10.1, pp. 288–299.
10. J. A. Snyman, Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms. New York: Springer-Verlag, 2005.
11. J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," in Proc. Of the IEEE Int. Conf. on Neural Networks. Piscataway, NJ: IEEE Service Center, 1995, pp. 1942–1948.
12. Sambarta Dasgupta, Swagatam Das, Ajith Abraham, Adaptive Computational Chemotaxis in Bacterial Foraging Optimization: An Analysis, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 13, NO. 4, AUGUST 2009
13. Qiaoling Wang, Xiao-Zhi Gao and Changhong Wang, AN ADAPTIVE BACTERIAL FORAGING ALGORITHM FOR CONSTRAINED OPTIMIZATION, International Journal of Innovative Computing, Information and Control Volume 6, Number 8, August 2010
14. Hai Shen, Yunlong Zhu, Xiaoming Zhou, Haifeng Guo, Chunguang Chang, Bacterial Foraging Optimization Algorithm with Particle Swarm Optimization Strategy for Global Numerical Optimization, GEC'09, June 12–14, 2009, Shanghai, China.

#### AUTHOR PROFILE

**Mohd. Ilyas** is Asst. professor in the Department of Electrical & Electronics Engineering in Al-Falah School of Engineering And Technology. He is having twelve years of professional experience. He did his graduation in Electrical Engineering. In 1999 and M.tech in Electrical Power System Management from Jamia Millia Islamia University, New Delhi in 2010 and now perusing PH D from Al-Falah School of Engineering And Technology(M.D.U). He has attended many National and International seminars and workshops. He started his Teaching career in 2001 as a Lecturer in AFSET, Part time Faculty in G.B. Pant Polytechnic

last 5 years. He published Several Papers in national and international Journals. His area of interest is Electrical Engineering.

**Hema Sharma** is student of M.tech (Power system) 4<sup>th</sup> sem in Al-Falah School of Engineering And Technology . She did her B.tech in Electrical & Electronics Engineering from LIMAT Faridabad in 2010.

**Surakant Shukla** is Asst. Professor in HMRITM Delhi. He did his B.Tech in Electrical & Electronics Engineering from BBIET & RC and M.Tech in Electrical Engineering from NIT Hamirpur. He published Twelve Papers in national and international conferences & journals.