Comparison of the PSO Algorithm and JAYA Algorithm for Economic Load Dispatch

S. Nagaraju[1], B. Ravi Kumar[2], Reddi Hari Chandana[3], PSV Satya Sai[4], Jonnada Sirisha[5], Lolla Kishore Kumar[6]

Associate professor[1], Department of Electrical and Electronics Engineering,
Assistant professor[2], Department of Electrical and Electronics Engineering
3,4,5,6 students department of Electrical and Electronics Engineering, Aditya Institute of Technology and Management, Tekkali, India.

---***---

Abstract – The electrical power industry has undergone the significant changes, leading to a competitive market and the need for optimal economic dispatch solutions. As energy resources becoming scarce, prices rise, environmental concerns intensify, and electricity consumption increases, classical optimization techniques struggle to handle these complex issues. This project compares particle swarm optimization (PSO) and JAYA optimization method for economic load dispatch (ELD) problems in the electrical power industry. PSO has gained recognition as an effective solution for ELD problems in the last decade. The aim of this study is to show that Jaya optimization is the best optimization methodology for ELD problems. By comparing the performance, convergence speed, and accuracy of PSO and JAYA optimization, this paper aims to provide insights into the superiority of JAYA optimization.

***Key Words*:** Economic Load dispatch, Optimization method, PSO algorithm, JAYA algorithm, Incremental fuel cost

1.INTRODUCTION

Economic load dispatch (ELD) is a critical aspect of power system management, where multiple units produce more electricity than needed. This extra power generation necessitates efficient distribution among committed generators. However, geographic distance between power plants and load centers poses a significant challenge, impacting operational issues and fuel prices. Traditional scheduling techniques like the gradient approach, base point and participation factor method, and lambda iteration method are used to manage load demand while conserving costs. However, these methods have limitations in managing the complexity of real-world power systems. Modern power systems consist of complex networks of linked units, presenting challenges like multi-fuel alternatives, ramp rate limitations, forbidden operation zones, and valve-point loading. ELD is a non-convex optimization problem with discontinuities and non-liner features, necessitating the use of complex optimization methods.

Let us assume that C_i is the cost of energy production in generator unit I, represented in dollars per megawatt hour or rupees per megawatt hour. The entire cost of system manufacture is shown as

$$C = \sum_{i=1}^{n} \frac{c(i)\$}{h}$$

The prime mover torque increases the actual generation and fuel costs, but it does not significantly affect the generating unit's production cost, which is controlled by field current. Therefore, the overall controlled production cost of unit 1 is a function of P_{Gi}.

$$C = \sum_{i=1}^{n} C_i\,(P_{Gi})$$

System constraints are physical, operational, and regulatory requirements for the safe, reliable, and effective functioning of the power grid. They guide decision-making and maximize resource use. There are two main categories of constraints:

Equality Constraints

Within generation restrictions, equality constraints require that the generated power equals the total of the system's power demand (PD) and losses (PL). to maintain supply and demand balance, this equation makes sure that the power produced by the generators equals the overall demand as well as any losses in the power system.

$$P_D + P_L = \sum\nolimits_{i=1}^{n} P_{Gi}$$

Inequality Constraints

i. Generator constraints

Generator output limits, P_{min} and P_{max} , are determined by technical capabilities, operational limitations, and external conditions. Breaking these limits can cause system instability, so managing output is crucial.

$$P_{min} \le P \le P_{max}$$

Voltage constraints

The power system's nodes and buses must maintain voltage levels within designated ranges to prevent equipment malfunctions, unstable voltage, and potential voltage collapse.

Transmission Line Capacity Limits

The capacity of transmission lines is the maximum power flow that can be safely transferred, and exceeding this limit can lead to overloading, losses, equipment heat stress, and potential outages.

Lambda iteration method

The Lambda iteration method is a numerical method used to solve nonlinear equations repeatedly, particularly in complex cases where analytical solutions are impractical. It involves introducing a parameter, λ, into the original equation, updating it until convergence, and modifying it after each iteration to represent one variable in terms of others. The Lambda iteration approach is an iterative method that prompts convergence in solving complex or implicit equations. It excels in addressing numerical problems in fields like engineering, physics, economics, and optimization. The approach moves across the solution space, adjusting λ values at each iteration to improve convergence and accuracy. It can handle real-world systems complexity and non-linearities, making it a useful tool for numerical problems. It can also be implemented computer-based for automatic equation solving that defy analytical resolution.

The Lambda iteration method is a useful tool for solving complex nonlinear equations in various domains, advancing scientific study and engineering practice with the right tuning and parameter selection, despite some convergence behavior restrictions and the influence of the equation's properties and λ selection.

ALGORITHM DESCRIPTION

Step 1: Read the given data

Step 2: Choose the initial value of λ and Δλ.

Step 3: Determine P_{gi} corresponding to incremental fuel cost.

Step 4: For each unit check the generation limits.

If $P_{gi} > P_{gi}$ max set $P_{gi} = P_{gi}$max

If $P_{gi} < P_{gi}$ min set $P_{gi} = P_{gi}$min

Step 5: the difference in power at all generators bus between consecutive iteration should be less than prescribed value. If not, go to step 3.

Step 6: After all, P_{Gi} values are calculated, find out the loss using equation (3). Calculate the mismatch between generator power and demand including losses.

ΔP= $\sum_{i=0}^{NG} P_{Gi} - P_D$

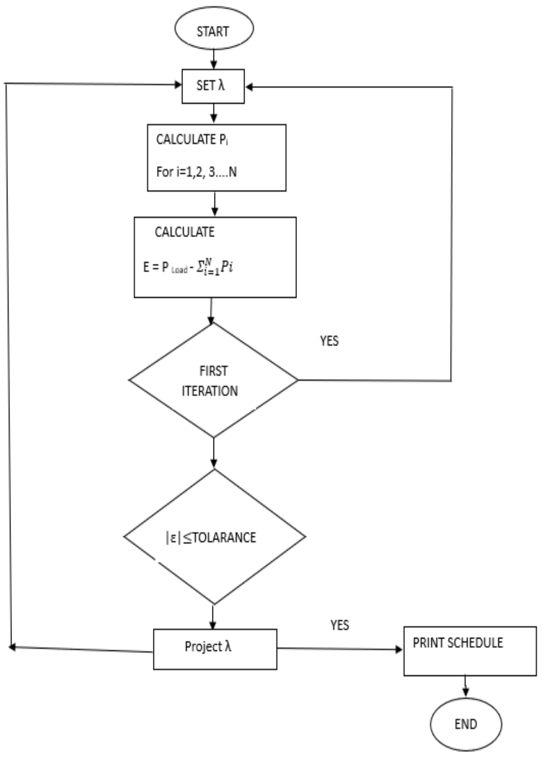
Step 7: If the value of ΔP is the less than some specified value then stop calculation and calculate cost of generation with these values of power. Otherwise go to step 8.

Step 8: increase the value of λ & Δλ; if ΔP < 0 or

Decrease the value of λ & Δλ; if ΔP < 0

And repeat from step 4

FLOW CHART



PARTICLE SWARM OPTIMIZATION

Particle swarm optimization is a population-based stochastic search technique in combinatorial metaheuristic optimization. It addresses large-scale issues like differentiability, dimensionality, and multimodality. Conventional techniques like dynamic programming and linear programming struggle with nonlinear goal functions. Stronger optimization techniques are needed to overcome these problems. The economical load dispatch problem is a crucial power system issue, often addresses using Lagrange multipliers and monotonically increasing piecewise linear cost functions. PSO is a computer method that optimizes problems by continuously improving a candidate solution based on a quality measure. It uses a population of possible solutions, called particles, and adjusts them in the search space using a mathematical formula. The particle's position affects its mobility and it is guided towards the best repair locations.

So, what is the most effective way to find the food?

The bird's search for food involves adjusting its position and velocity based on past encounters and neighboring remarks. This optimization technique, inspired by fish schools and bird flocks, can be used to address non-linear optimization issues. PSO uses a multitude of solutions, or particles, to create a swarm to find the best answer, ensuring a more efficient search process. The search space describes a particle's position and velocity using X and V, with each particle represented by Xi. Iteration is used to update each particle with the two best values.

Particles log their personal best (p best) and neighborhood best (G beat) values, and when they identify these ideal values, they modify their positions and speed. The global best particle (g best) performs best among all particles in the multidimensional search space. The modified position and velocity of each particle are determined based on current velocity and distance from pbest and gbest.

$$\boldsymbol{v_i^{(t+1)} = w * v_i^t + C1 * rand(\,) * (pbest_i - p_i^t) + C2 * rand(\,) * (gbest_i - p_i^t)}$$

$$\boldsymbol{P_i^{(t+1)} = p_i^t + V_i^{(t+1)}}$$

Algorithm

The PSO algorithm is used to identify the optimal distribution of electricity using committed units to reduce generating costs. It considers every restriction for every producing unit. To address the ELD problem

Step 1: The algorithm starts with individual cost functions of numerous stations.

Step 2: set up population size, C1 and C2, Wmax and Wmin error gradients, and other parameters.

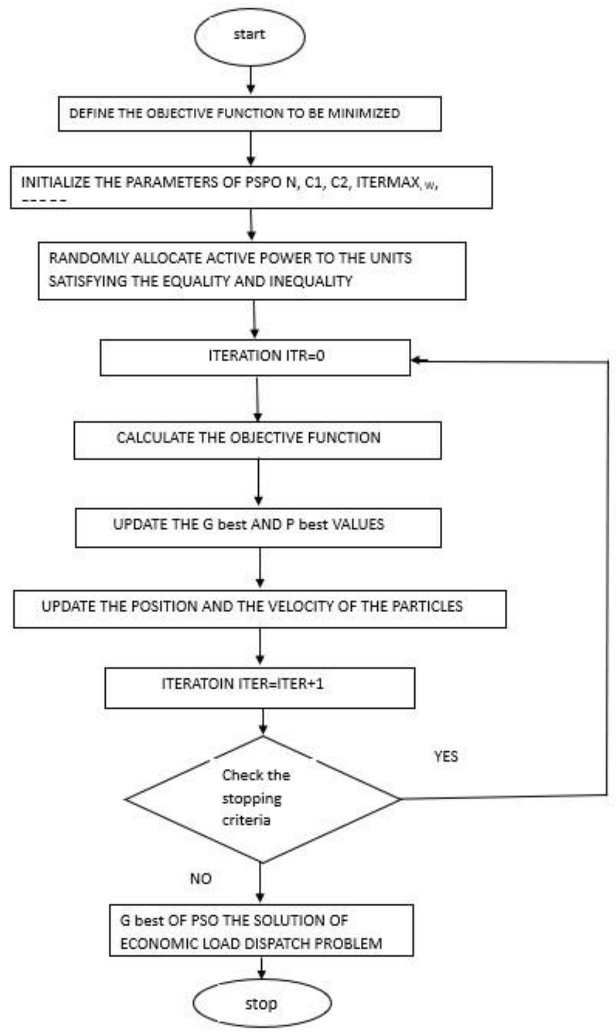
Step 3: Enter fuel cost functions, overall power demand, B-coefficients matrix, and MW, limitations of the producing stations.

Step 4: The program uses many randomly assigned active power vectors that meet the MW restrictions.

Step 5: The fitness function value is computed for each vector, and the Pbest and Gbest values are determined by comparing values from each iteration.

Step 6: The error gradient is examined at each step, and the best value for G is shown until it falls within a predetermined range.

FLOW CHART of PSO [3]


start
DEFINE THE OBJECTIVE FUNCTION TO BE MINIMIZED
INITIALIZE THE PARAMETERS OF PSPO N, C1, C2, ITERMAX, w, -----
RANDOMLY ALLOCATE ACTIVE POWER TO THE UNITS SATISFYING THE EQUALITY AND INEQUALITY
ITERATION ITR=0
CALCULATE THE OBJECTIVE FUNCTION
UPDATE THE G best AND P best VALUES
UPDATE THE POSITION AND THE VELOCITY OF THE PARTICLES
ITERATOIN ITER=ITER+1
Check the stopping criteria
YES
NO
G best OF PSO THE SOLUTION OF ECONOMIC LOAD DISPATCH PROBLEM
stop

JAYA ALGORITHM

Jaya Algorithm is a Population based technique, combines the global optimum solution appeal of swarm intelligence techniques with the survival of the fittest concept from evolutionary algorithms. It can be applied to the function minimization or maximization, continuously modifying a population of individual solutions. It is introduced in 2016, a user-friendly, straightforward, comprehensive, versatile, and the adaptive algorithm widely used in the various optimization problems, including solar cell parameter extraction and the optimal power flow. Researchers have enhanced the convergence behavior of the JAYA method, blending It with other optimization techniques to address search space challenges and the optimization issues. This parameter-less, globally effective techniques are used for both restricted and unconstrained benchmark functions. It is aims to solve the limited and unconstrained optimization functions using a population-based metaheuristic. This superior method offers ease of use and focus on survival

Algorithm:

Step1: The optimization problem's initial parameters, population size N and iteration numbers T, are set during the first run stage in f(x)

s.t.

gj (x) = c_j where j = (1,2,3…. n)

hk (x) = d_k where k = (1,2,3….n)

where x_i is a decision variable assigned a value in the lower and upper bound range such than xi ∈ [x_I min, $x_i$max], and f(x) is the objective function used to determine the fitness value of the solution x = (x1,x2,….,xD). HK represents the k inequality constraints, and gj represents the jth constraints. Typically, a benchmark dataset's dimensions, issue variables, and associated data are extracted.

Step 2: Building the JAYA starting population, the JAYA memory (JM) is where the initial solutions, or population, of the JAYA algorithm are created and stored. As indicated, where N is the number of solutions and D is the solution dimension, the JM is an augmented matrix of size N×D. Typically, the answer is generated at random using the formula $JM_{i,j}$ = XJ_{min} + (XJ_{min} - XJ_{max})×rand , where i ∈ (1,2,….N) and j ∈ $(1,2,\ldots\ldots N)$. The uniform function, rnd, produces a random number between 0 and 1

$$JM=\begin{bmatrix} x_1^1 & x_2^1 & \dots & x_D^1 \\ x_1^2 & x_2^2 & \dots & x_D^2 \\ \vdots & \vdots & \dots & \vdots \\ x_1^N & x_2^N & \dots & x_D^N \end{bmatrix}\begin{bmatrix} f(x^1) \\ f(x^2) \\ \vdots \\ f(x^N) \end{bmatrix}$$

The JM solutions are ranked based on their objective functions, with × 1 being the ideal and × N being the worst.

Step 3: The JAYA Evolution procedure have the decision variable changed iteration by iteration utilizing the JAYA operator

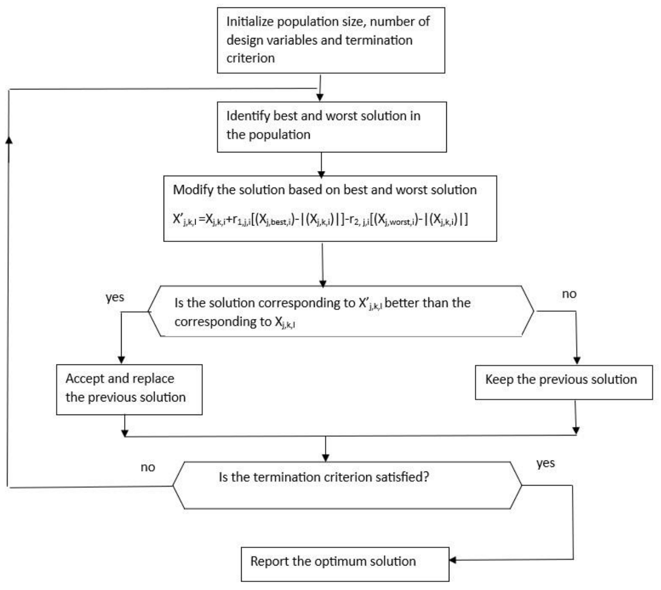
$$x_j^i = x_j^i + r_1 \times (x_j^1 - |x_j^i|) - r_2 \times (x_j^N - |x_j^i|)$$

The current X_{ij}, which is the newly updated version. The modified value of the decision values, x_j^i is $x_{j,}$ i. The two uniform functions r1 and r2, produce the random result between 0 and 1. To achieve the ideal balance between the exploration and exploitation phases, these randomly generated numbers are employed. It should be noted that the decision variable J in the best solution is x_j^1, while in the worst is x_j^N. The JAYA algorithm diversity control is determined by the separation between the decision variables of the current solution, as well as the separation between the decision variables of the worst solutions and the current one.

Step 4: There will be updates to the JM solutions at each phase. The new solutions $f(x^j)$ is objective function value is computed. If $f(x)<f(x^i)$, the new solution x^i will take the place of the existing solution x^i. The number of times this process is performed is N.

Step 5: Stop rule. The JAYA algorithm repeats step3 and step 4, until the stopping rule which is sometimes the maximum number of iterations T is reached.[1]

Flow Chart of JAYA Algorithm [2]


Initialize population size, number of design variables and termination criterion
Identify best and worst solution in the population
Modify the solution based on best and worst solution
$X'_{j,k,l} = X_{j,k,i} + r_{1,j,i}[(X_{j,best,i}) - |(X_{j,k,i})|] - r_{2,j,i}[(X_{j,worst,i}) - |(X_{j,k,i})|]$
yes
Is the solution corresponding to $X'_{j,k,l}$ better than the corresponding to $X_{j,k,l}$
no
Accept and replace the previous solution
Keep the previous solution
no
Is the termination criterion satisfied?
yes
Report the optimum solution

Results:

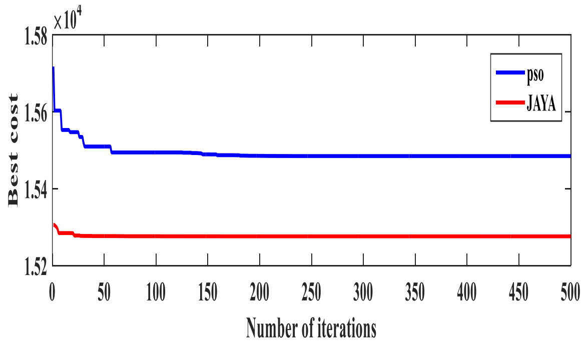
Data for 3 UNIT system with demand of 850MW

Generator	P_{min} (MW)	P_{max} (MW)	a	b	c	e	f
1	100	600	0.001562	7.92	561	300	0.0315
2	50	200	0.004820	7.97	78	150	0.063
3	100	400	0.001940	7.85	310	200	0.042

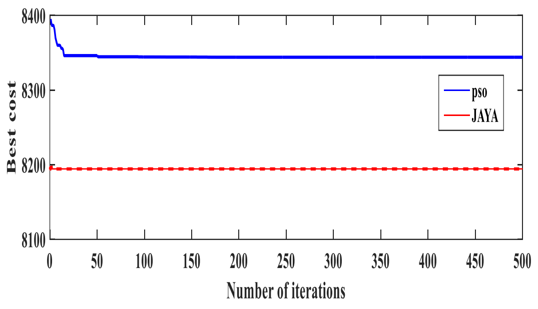
Table 1: Input data for 3-unit system for 850MW

Sl.no	PSO Algorithm	JAYA Algorithm
P_{g1}	498.9324	433.9110
P_{g2}	99.8666	191.4896
P_{g3}	251.2010	224.5994
Fuel cost (Rs/hr)	8411.2	8194.4

Table 2: Test results for comparison of PSO and JAYA algorithm



Comparison of characteristics of PSO and JAYA algorithm



Comparison of characteristics of PSO and JAYA algorithm

Generator	P_{min} (MW)	P_{max} (MW)	a	b	c	e	f
1	100	600	0.001562	7.92	561	300	0.0315
2	50	200	0.004820	7.97	78	150	0.063
3	100	400	0.001940	7.85	310	200	0.042

Table 6.5: Input data for 3-unit system with losses for 418 MW

$$B=\begin{bmatrix} 0.000136 & 0.0000175 & 0.000184 \\ 0.0000175 & 0.000154 & 0.000283 \\ 0.000184 & 0.000283 & 0.00116 \end{bmatrix}$$

$B_0 = [0.0046 \quad 0.0035 \quad 0.0019]$

$B_{00} = 0.00055711$;

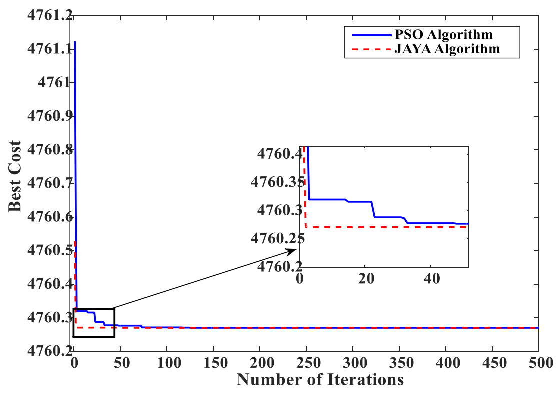
Sl.no	P_{min}	P_{max}	a	b	c	e	f
1	100	500	0.007	7	240	300	0.031
2	50	200	0.009	10	200	150	0.063
3	80	300	0.009	8.5	220	200	0.042
4	50	150	0.009	11	200	100	0.08
5	50	200	0.008	10.5	220	150	0.063
6	50	120	0.0075	12	190	100	0.084

Table: Input data for 6-unit system for demand of 1263 MW

Sl.no	PSO Algorithm	JAYA Algorithm
P_{g1}	206.8185	184.0265
P_{g2}	85.8899	45.5479
P_{g3}	18.8542	70.4256
Fuelcost (Rs/hr)	4779.9	4760.8
Total losses	10.6273	18.6972

Sl.no	PSO Algorithm	JAYA algorithm
1	500	447.1385
2	149.7999	171.1681
3	231.7365	263.7089
4	87.2572	125.4657
5	200	172.2345
6	90.2505	83.2844
FuelCost (Rs/hr)	15459	15276

Table: Test results for comparison of PSO algorithm and JAYA algorithm without losses



Cocomparison of characteristics of PSO and JAYA algorithm

Sl.no	P_{min}	P_{max}	a	b	c	e	f
1	100	500	0.007	7	240	300	0.031
2	50	200	0.009	10	200	150	0.063
3	80	300	0.009	8.5	220	200	0.042
4	50	150	0.009	11	200	100	0.08
5	50	200	0.008	10.5	220	150	0.063
6	50	120	0.0075	12	190	100	0.084

Table: Input data for 6-unit system with losses for demand of 1263 MW

B = [0.0017 0.0012 0.0007 -0.001 -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;

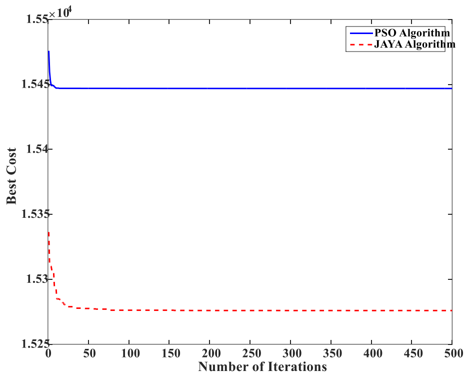
-0.0002 -0.0001 -0.0006 -0.0008 -0.0002 0.0150];

B0 = 1.0e-03*[-0.3908 -0.1297 0.7047 0.0591 0.2161 -0.6635];

B00 = 0.0056;

Sl.no	PSO Algorithm	JAYA algorithm
1	447.80101	446.6700
2	172.1911	171.1822
3	264.4727	264.2278
4	139.6207	125.8430
5	164.6822	171.8836
6	86.9313	83.1932
FuelCost (Rs/hr)	15444.8	15334
Total loss	12.776	12.9162

Table 6.8: comparison of PSO and JAYA algorithm with losses



Comparison of characteristics of PSO and JAYA algorithm

CONCLUSION

In this project, a classic Jaya algorithm is implemented to solve the ELD problem for the three- and six-unit power system. The feasibility of the Jaya Algorithm is verified by comparing with the PSO algorithm. The comparison of the results for the test cases of three units and six units power system clearly shows that the proposed method is indeed capable of obtaining quality solution efficiently for the different systems of ELD problems. The convergence characteristics of the proposed algorithm for the three- and six-unit systems are plotted. Thus, ELD solution for the different system units can be obtained with optimum cost of generation from JAYA OPTIMIZATION than the conventional methods.

REFERENCES:

[1] Zitar, R. A., Al-Betar, M. A., Awadallah, M. A., Doush, I. A., & Assaleh, K. (2021). An Intensive and Comprehensive Overview of JAYA Algorithm, its Versions and Applications. Archives of Computational Methods in Engineering, 29(2), 763–792. https://doi.org/10.1007/s11831-021-09585-8

[2] Banerjee, S., & Sarkar, D. (2018). Modified Jaya Optimization Algorithm for Combined Economic Emission Dispatch Solution. International Journal of Electrical Energy, 6(1), 13–19. https://doi.org/10.18178/ijoee.6.1.13-19

[3] Safari, A., & Shayeghi, H. (2011). Iteration particle swarm optimization procedure for economic load dispatch with generator constraints. Expert Systems With Applications, 38(5), 6043–6048. https://doi.org/10.1016/j.eswa.2010.11.015

[4] Sivanagaraju, S. (2009). Power System Operation and Control. Pearson Education India

[5] Rao R (2016) Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int J Indus Eng Comput 7(1):19–3

[6] Jan, Z. N. (2021). Economic Load Dispatch using Lambda Iteration, Particle Swarm Optimization &amp; Genetic Algorithm. In International Journal for Research in Applied Science and Engineering Technology (Vol. 9, Issue 8, pp. 972–977). International Journal for Research in Applied Science and Engineering Technology (IJRASET). https://doi.org/10.22214/ijraset.2021.37527 - reference paper