IJCRT.ORG ISSN : 2320-2882
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

MINIMISATION OF COST OF GENERATION USING PARTICLE BEVY OPTIMIZATION

Vivek Viswanadha
Vignan's Institute of Information Technology

ABSTRACT— Economical load dispatch is process of real power setting on generators with an objective of minimizing fuel cost while satisfying equality and inequality constraints. Economic load dispatch problem is solved by classical manual and computational methods which takes long time to reach optimal solution. In practical there are many generating units so it is requisite to have an optimization method which provides optimal solution in less time. Particle bevy optimization algorithm is applied to economical load dispatch is capable of providing faster response and higher convergence rate.
.

Index Terms— Economic load dispatch, particle bevy optimization, objective function,
Position, velocity.

I. Introduction

PARTICLE BEVY OPTIMIZATION

Particle bevy optimization (PSO) is population based stochastic optimization technique developed by Dr. Kennedy and Dr. Eberhart in 1995, inspired by social behavior of bird flocking. In PSO, the solutions are called as particles, these fly in search space of problem until an optimal solution is reached or stopping criteria is met. Each particle keeps tracking it's coordinates under the control variables they are

(i) Velocity
(ii) Personal best (P_{Best})
(iii) Global best (G_{Best})
(iv) Target value

Let S be the size of the bevy, each i^{th} particle can be indicated as an thing with many characteristics. A population of particles is initialized with arbitary position X_i and velocities V_i objective function F_i is calculated using the particles positional coordinates as input values. This value is called P_{Best}. Another best value that is tracked by the global version of the bevy is the overall best value, and its place obtained so far by any particle in the population. This place is called G_{Best}. At each time velocity of each particle flying toward its G_{Best} and P_{Best} place is changed. Rate of chane of velocity is weighted by arbitary terms, with separate arbitary numbers being generated for the rate towards P_{Best} and G_{Best} location. At each time position and rate of position are adjusted and the function is modified with new coordinates. When the particle discovers a solution better than the previous one, it stores the coordinates in the vector $P_{Best\ id}$. The difference between the best point found by a particular agent and the individual's current positions is added to the current velocity causing the trajectory to oscillate around the point. Further each particle is defined within the context of a topological neighborhood comprising itself and some other particles in the population. The stochastically weighted difference between the neighborhood's best position $G_{Best\ id}$ and the individual's current position is also added to its velocity, adjusting it for the next time step. These adjustments to the particle's movement through the space cause it to search around the two best positions.

Particle (X): It is the candidate solution indicated by a d – dimensional vector.

Where d is the number of optimized parameters. At time t, the i^{th}

particle $X_i(t)$ is given by $X_i(t) = [X_{i1}(t), X_{i2}(t) \ldots X_{id}(t)]$.

Population X(t): It is the set of n particles at time t i.e $X(t) = [X_1(t), X_2(t) \ldots X_n(t)]$.

Bevy: It is an apparently disorganized population of moving particles that tend to cluster together while each particle seems to be moving in a arbitary direction. Particle velocity V(t): It is the velocity of the moving particles indicated by a d- dimensional vector. At time t, the i^{th} particle velocity $V_i(t)$ can be described as $V_i(t) = [V_{i1}(t), V_{i2}(t) \ldots V_{id}(t)]$, where V_{id} is the velocity of i^{th} particle with respect to the d^{th} dimension. The velocity update step is specified separately for each dimension d = 1…n, so that V_{id} denotes the d^{th} dimension of the velocity vector associated with the i^{th} particle.

Velocity updation is given by eqn:

$$V_{id}^{(t+1)} = wV_{id}^{(t)} + C_1 * rand_1 (P_{Best\,id}^{(t)} - X_{id}^{(t)}) + C_2 * rand_2 (G_{Best\,id}^{(t)} - X_{id}^{(t)}).$$

Where

$V_{id}^{(t)}$, $X_{id}^{(t)}$ are velocity and position of particle.

$P_{Best\,id}^{(t)}$ & $G_{Best\,id}^{(t)}$ are personal and global best positions.

W is the weight inertia.

C_1 & C_2 are accelerating constants, $rand_1$ & $rand_2$ are arbitary variables in the range [0, 1].

The velocity of each particle should be clamped between $[-V_{max}, V_{max}]$ to reduce the chances of particle leaving the search space.

Particle position updating eqn:

$$X_{id}^{(t+1)} = X_{id}^{(t)} + V_{id}^{(t+1)}$$

Inertia weight (w): The inertia weight controls the exploration and exploitation of the search space because it dynamically adjusts velocity. The inertia weight is employed to control the effect of the previous velocities on the current velocity. This makes compromise between a global and (wide ranging) and local (nearby) exploration abilities of the bevy. A large inertia weight facilitates global exploration (searching new areas) while a small one tend to facilitate local exploration. A properly chosen inertia weight can provide balance between the global and local exploration of the bevy, which leads to a better solution. It is better to initially set the inertia weight to a large value in order to make better global exploration of the search space and gradually decrease it to get more refined solution. A linearly decreasing inertia weight changes the search from global to local linearly.

$$W = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} * iter$$

Where w_{max} is the maximum value of inertia weight and w_{min} is the minimum value, iter is the current iteration and $iter_{max}$ is the maximum number of iterations.

Accelerating coefficients (C_1, C_2): The accelerating coefficients C_1&C_2 represent the weighting of the cognitive and social parameters that pulls each particle towards P_{Best} and G_{Best} positions.

Personal best: The personal best position associated with the i^{th} particle is the best position that the particle has visited

$$P_{Best\,id}^{(t+1)} = \begin{cases} X_{id}^{(t)} & \text{if } f\left(X_{id}^{(t+1)}\right) \geq f(P_{Best\,id}^{(t)}) \\ X_{id}^{(t+1)} & \text{if } f\left(X_{id}^{(t+1)}\right) < f(P_{Best\,id}^{(t)}) \end{cases}$$

Global best: The best solution of all the personal best solutions in the entire bevy.

2.Search Mechanism of PSO:

Figure shows the search mechanism of PSO. Each particle moves from the current position to the next one according to the present fitness function values. Generally, the fitness function is same the objective functions. The local best of other particles in the population should be changed if the present fitness function value is better than the previous. Repeat the new searching points until the maximum number of generations reached. 150 generations are set in this paper as the stopping criteria.

More importantly, every particle has a memory capacity, and can provide one-way message to the population. Thus, the search process of PSO is the process of following current optimal solution. For example, if food distance is known to the population but place is unknown, the simplest way to find the food is to search the peripheral regions of the birds that are closest to the food. The solution program first sets the end condition (number of iterations or error tolerance), and obtains the optimal solution lastly

Each particle keeps track of its coordinates in the problem space, which are associated with the best solution, fitness, it has achieved so far. The fitness value is also stored. This value is called pbest. Another best value that is tracked by the particle bevy optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This place is called lbest.When a particle takes all the population as its topological neighbors, the best value is a global best and is called gbest.

In a PSO algorithm, particles change their positions by flying around in a multidimensional search space until a relatively unchanged position has been encountered, or until computational limitations are exceeded. In social science context, a PSO system combines a social-only model and a cognition-only model. The social-only component suggests that individuals ignore their own experience and fine tune their behavior according to the successful beliefs of the individual in the neighborhood. On the other hand, the cognition-only component treats individuals as isolated beings. A particle changes its position using these models.

The particles continue flying and seeking solution and hence the algorithm continues until a pre specified numbers of maximum iterations are exceeded or exit criteria are met. The accuracy and rate of convergence of the algorithm depends on the appropriate choice of particle size, maximum velocity of particles and the inertia weight. However, no specific guideline is available to select the particle size. Moreover, it also varies from problem to problem. As a result, one has to choose it by trial and error. The maximum velocity of individual particles should be chosen very judiciously.

PSO parameters used for simulation:

Parameter	PSO
Population size	20
Number of iterations	150
Cognitive constant, c1	2
Social constant, c2	2
Inertia weight, W	0.3-0.95

Single Objective function:

The objective of OPF selected is minimization of cost of generation. The optimization problem is mathematically given in equation.

$$\text{Minimize} \quad F = \sum_{i=1}^{ng} (a_i P_{gi}^2 + b_i P_{gi} + c_i)$$

Where

P_{gi} is the amount of generations in MW at generator i.

a_i , b_i , c_i are the cost coefficients.

ng = number of generators including slack bus.

To minimize $F(x)$ is equivalent to getting a maximum fitness value in the searching process. The particle that has a lower value of the function should be assigned a larger fitness value.

The objective of OPF has to be changed to the maximization of fitness to be used as follows

$$\text{fitness} = f_{max} = 1/F(x)$$

Where

F = Fuel cost minimization

Equality constraints: In equality constraints, the generating power should match the load demand. It is given by equation

$$\sum_{i=1}^{ng} P_{gi} - P_d = 0$$

Inequality constraints: For each generating unit the power generation must be within the maximum and minimum limits.

$$P_{gi(min)} \leq P_{gi} \leq P_{gi(max)}$$

PSO FLOW CHART

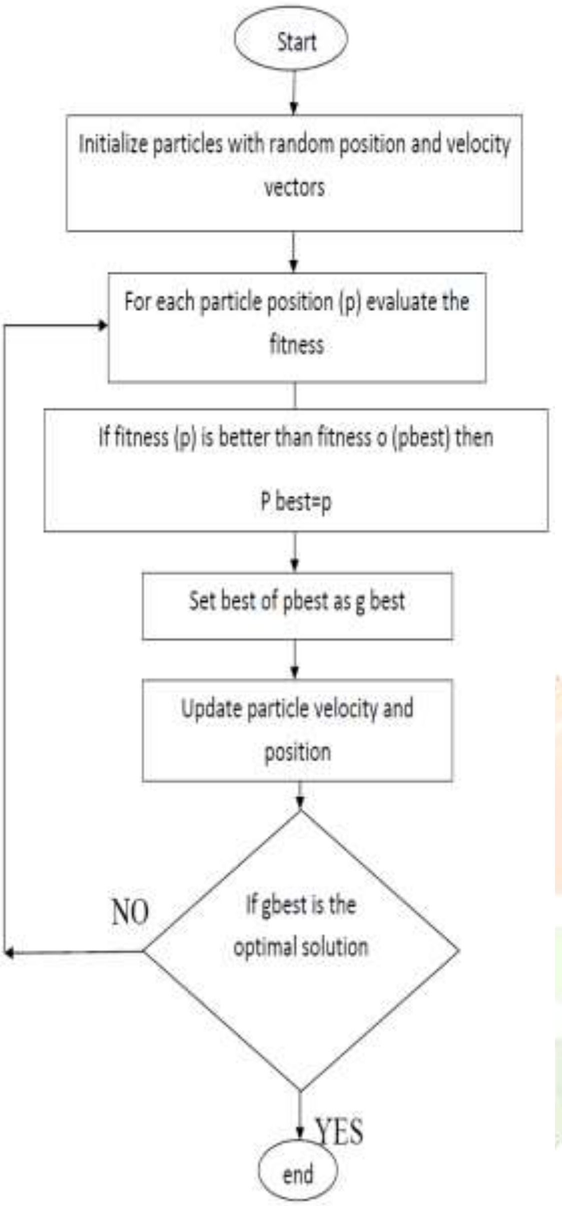


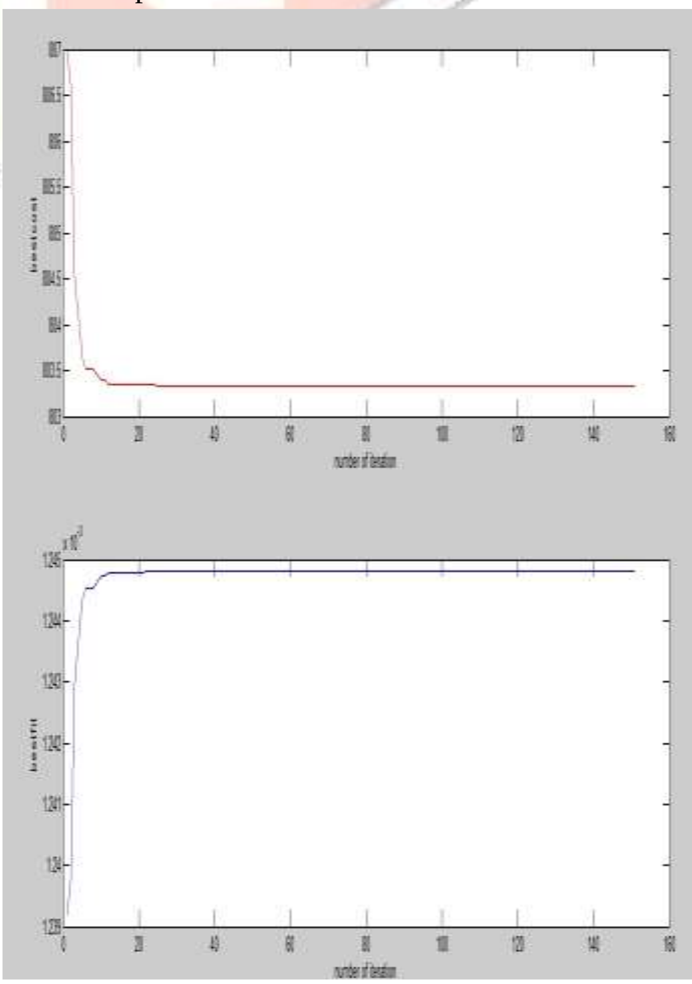
Fig: Flow chart of PARTICLE BEVY OPTIMIZATION

4. Computational Procedure:

Step 1: Read the input data of system.
Step 2: Initialize the population and PSO control variables.
Step 3: Initialize the maximum, minimum limits of velocity for each particle.
Step 4: Arbitaryly generate the velocities and populations.
Step 5: Update the system data according to the population generated.
Step 6: Compute the objective function and fitness values.
Step 7: Repeat the step5 and step6 for all the populations and select the best fitness value as global fit value and corresponding particles are Gbest values.
Step 8: Initialize the iteration counter and start the iteration process.
Step 9: Update the velocities and positions values, check the updated velocities lies within limits or not. Fix those values minimum or maximum according to their violation.
Step 10: Repeat the steps from step5 to step9 for all populations.
Step 11: Update local best and global best values.
Step 12: Repeat the steps from step9 to step11 until Iter ≤ IterMax.
Step 13: Stop the process and print the Gbest value.

5.SIMULATION RESULTS

A program for economic load dispatch using P.S.O has developed in the Matlab and results are obtained.



Number of Iterations V/s best cost, best fit

6.Conclusion

The work is applied in the area of economic load dispatch comes under single objective optimization. The above work is applied to the IEEE-6 bus system. Minimization of fuel cost achieved by using the particle bevy optimization. This work is carried out by considering the generator bus data and load demand

References

[1] “Power system operation and control” by siva nagaraju.

[2] Mahajan, M., & Vadhera, S. (2012). Economic load dispatch of different bus systems using particle bevy optimization.2012 IEEE 5th power India conference. doi:10.1109/poweri.2012.6479555

[3] Alam, M. N., Mathur, A., & Kumar, K. (2016). Economic load dispatch using a differential particle bevy optimization. 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems(ICPEICES). doi:10.1109/icpeices.2016.7853592.

[4] S. Sahoo, K. Dash, and A. Barisal, "Solution of economic load dispatch by evolutionary optimization algorithms: A comparative study," in IEEE International Conference on Control, Instrumentation, Energy and Communication (CIEC), Jan 2014, pp. 259-263.

[5] R. Eberhart and J Kennedy, "A new optimizer using partic1e bevy theory," in IEEE Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Oct 1995, pp. 39-43.