

“Optimal Power Flow Using Differential Evolution”

Tarabhan Gupta, Mrs. Aakansha Mercy Steele

Student M.Tech (Power System) UIT-RGPV BHOPAL

Assistant Professor UIT-RGPV BHOPAL

Corresponding Author: Tarabhan Gupta

ABSTRACT

The work deals with the optimization of power flow problem through interconnected system. The work presents an algorithm for solving optimal power flow problem through the application of Differential Evolution (DE). The objective is to minimize the total fuel cost of thermal generating units having quadratic cost characteristics subjected to limits on generator real and reactive power outputs, bus voltages, transformer taps and power flow of transmission lines. The work introduces a conceptual overview and detailed explanation of Differential Evolution algorithm as well as shows how it can be used for solving optimal power flow problems. Inherent shortcoming of the traditional methods of finding the optimal power solution is discussed along with other evolutionary methods of finding optimal power solution. A comparative study of different evolutionary programming technique is done and it is shown that Differential Evolution offer a better result with greater repeatability and lesser time.

The proposed method has been tested under simulated conditions on IEEE 30-bus system.

The optimal power flow results obtained using Differential Evolution are compared with other Genetic Algorithm. It is shown that Differential Evolution total generation fuel cost is less expensive than those of evolutionary programming, Gradient projection method (GPM), SLP, QN.

Keywords: Optimal Power Flow using Differential Evolution.

Date of Submission: 10-02-2018

Date of acceptance: 28-02-2018

I. INTRODUCTION

Optimization is the process of making something better. an engineer or scientist conjures up to a new idea and optimization improves on that idea. optimization consists in trying variations on an initial concept and using the information gathered to improve on the idea. Evolutionary algorithms (eas) are optimization techniques based on the concept of a population of individuals that evolve and improve their fitness through probabilistic operators like recombination and mutation. These individuals are evaluated and those that perform better are selected to compose the population in the next generation. After several generations these individuals improve their fitness as they explore the solution space for optimal value. The field of evolutionary computation has experienced significant growth in the optimization area. These algorithms are capable of solving complex optimization problems such as those with a non-continuous, non-convex and highly nonlinear solution space. In addition, they can solve problem that feature discrete or binary variables, which are extremely difficult. Several algorithms have been developed within the field of evolutionary computation (EC) being the most studied genetic algorithms were first conceived in

the 1960's when evolutionary computation started to get attention. Recently, the success achieved by EAs in the solution of complex problems and the improvement made in computation such as parallel computation have stimulated the development of new algorithms like differential evolution (DE), particle swarm optimization (PSO), ant colony optimization (ACO) and scatter search present great convergence characteristics and capability of determining global optima. Evolutionary algorithms have been successfully. Applied to many optimization problems within the power systems area and to the economic dispatch problem in particular [1-18].

Optimization techniques: Optimization algorithms have a very broad range of application, since many problems in Power System can be formulated as an optimization task where the objective is to minimize or maximize a given objective function. The different optimization techniques are.

Evolutionary Algorithms: Evolutionary algorithms are iterative and stochastic optimization techniques inspired by concepts from Darwinian evolution theory. An EA simulates an

evolutionary process on a population of individuals with the purpose of evolving the best possible approximate solution to the optimization problem at hand. In the simulation cycle, three operations are typically in play; recombination, mutation, and selection. Recombination and mutation create new candidate solutions, whereas selection weeds out the candidates with low fitness, which is evaluated by the objective function.

Simulated Annealing: The working principle of simulated annealing is borrowed from metallurgy: a piece of metal is heated and then the metal is left to cool slowly. The slow and regular cooling of the metal allows the atoms to slide progressively toward their most stable, minimal energy, positions. Rapid cooling would have "frozen" them in whatever position they happened to be at that time. The resulting structure of the metal is stronger and more stable.

Ant Colony Optimization: Ant systems, also known as an algorithms or ant colony optimizers, are multi-agent systems in which the behavior of each computer agent is inspired by the behavior of real ants. When real-world ant colonies are given access to a food source that has multiple approach paths, most ants end up using the shortest and most efficient route. The attempt to develop algorithms inspired by one aspect of ant behavior, the ability to find what computer scientists would call shortest paths, has become the field of ant colony optimization (ACO), the most successful and widely recognized algorithmic technique based on ant behavior.

Particle Swarm Optimization: Particle Swarm Optimization is a biologically inspired method of search and optimization developed in 1995 by Dr. Eberhart and Dr. Kennedy. Based on the social behaviors of birds flocking or fish schooling, this technique represents possible solutions as "particles" as they "fly" like a swarm through the solution space. Like a flock, the swarm gravitates towards the "leader", the current best-known solution, accelerating and turning as better solutions are found.

Differential Evolution: The differential Evolution algorithm (DE) is a population based algorithm like genetic algorithm using the similar operators; crossover, mutation and selection. The main difference in constructing better solutions is that genetic algorithms rely on crossover while DE relies on mutation operators.

This main operation is based on the differences of randomly sampled pairs of solutions in the population.

Tabu Search: The basic idea of tabu search is to explore the search space of all feasible solutions by a sequence of moves. However, to escape from locally optimal but not globally optimal solutions and to prevent cycling, some moves, at one particular iteration, are classified as forbidden or tabu. Tabu moves are derived from the short-term and long-term history of the sequence of moves.

Fuzzy Systems: Fuzzy systems consist of a variety of concepts and techniques for representing and inferring knowledge that is imprecise, uncertain, or unreliable. Fuzzy systems can create rules that use approximate or subjective values, as well as incomplete or ambiguous data. Statements such as "hot" or "expensive" are not immediately suitable for use by an algorithm. Fuzzy systems provide the means for handling such information by expressing logic with some carefully defined imprecision. Unlike traditional programs that use simple IF-THEN rules, fuzzy systems use statements that closely model the way people think so that these statements can be interpreted by computers as well as human experts.

Problem formulation of OPF: An Optimal Power Flow (OPF) function schedules the power system controls to optimize an objective function while satisfying a set of nonlinear equality and inequality constraints. The equality constraints are the conventional power flow equations; the inequality constraints are the limits on the control and operating variables of the system. Mathematically, the OPF can be formulated as a constrained nonlinear optimization problem.

There are many possible objectives for an OPF. Some commonly implemented objectives are:

- fuel or active power cost optimization,
- active power loss minimization,
- minimum control-shift,
- minimum voltage deviations from unity, and
- minimum number of controls rescheduled.

In fuel cost minimization, the outputs of all generators, their voltages, LTC transformer taps and LTC phase shifter angles, and switched capacitors and reactors are control variables.

Basic OPF Problem:-The OPF problem is an

optimization problem that determines the power output of each online generator that will result in a least cost system operating state. The OPF problem can be written in the following form:

$$\begin{aligned} &\text{Minimize} && f(x) \\ &\text{Subjected to:} && g(x) = 0 \\ &&& H(x) \leq 0 \end{aligned}$$

$f(x)$ is the objective function, $g(x)$ and $h(x)$ are respectively the set of equality and inequality constraints. x is the vector of control and state variables.

Cost function: The objective of the OPF is to minimize the total system cost by adjusting the power output of each of the generators connected to the grid. The total system cost is modeled as the sum of the cost function of each generator (1). The generator cost curves are modeled with smooth quadratic functions given by:

$$f(x) = \sum_{i=1}^{n_g} a_i + b_i P_{gi} + c_i P_{gi}^2$$

where n_g is the number of online thermal units, P_{gi} is the active power generation at unit i and a_i , b_i and c_i are the cost coefficients of the i^{th} generator.

Equality constraints: The equality constraint is represented by the power balance constraint that reduces the power system to a basic principle of equilibrium between total system generation and total system loads. Equilibrium is only met when the total system generation (P_g) equals the total system load (P_D) plus system losses (P_L).

$$\sum_{i=1}^{n_g} P_{gi} - P_D - P_L = 0$$

The exact value of the system losses can only be determined by means of a power flow solution. The most popular approach for finding an approximate value of the losses is by way of Kron's loss formula (14), which approximates the losses as a function of the output level of the system generators.

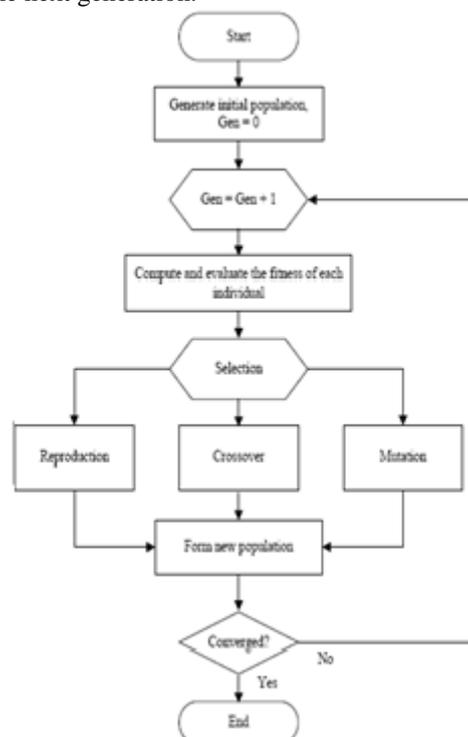
$$P_L = \sum_{i=1}^{n_g} P_{gi} B_{ij} P_{gj} + \sum_{i=1}^{n_g} P_{gi} B_{io} + B_{oo}$$

Differential Evolution (DE): -One extremely powerful algorithm from evolutionary computation due to its excellent convergence characteristics and few control parameters is differential evolution. Differential evolution solves real valued problems based on the principles of natural

evolution using a population P of N_p floating point-encoded individuals (1) that evolve over G generations to reach an optimal solution. In differential Evolution, the population size remains constant throughout the optimization process. Each individual or candidate solution is a vector that contains as many parameters (2) as the problem decision variables D . The basic strategy employs the difference of two randomly selected parameter vectors as the source of random variations for a third parameter vector. In the following, we present a more rigorous description of this new optimization method.

$$\begin{aligned} P &= [Y_1^{(G)} \dots\dots\dots Y_{Np}^{(G)}] \\ Y_i^{(G)} &= [X_{1i}^{(G)}, \dots\dots\dots X_{Di}^{(G)}] \\ &I = 1, 2, \dots\dots Np \end{aligned}$$

Extracting distance and direction information from the population to generate random deviations result in an adaptive scheme with excellent convergence properties. Differential Evolution creates new offsprings by generating a noisy replica of each individual of the population. The individual that performs better from the parent vector (target) and replica (trail vector) advances to the next generation.



Flow Chart For Differential Evolution

Control Parameter Selection: Proper selection of control parameters is very important for algorithm success and performance. The optimal control parameters are problem specific. Therefore, the set of control parameters that best fit each

problem have to be chosen carefully. The most common method used to select control parameters is parameter tuning. Parameter tuning adjusts the control parameters through testing until the best settings are determined. Typically, the following ranges are good initial estimates: F = [0.5], CR = [0.99], and NP = [20]. In order to avoid premature convergence, F or NP should be increased, or CR should be decreased. Larger values of F result in larger perturbations and better probabilities to escape from local optima, while lower CR preserves more diversity in the population thus avoiding local optima.

Constraint Handling:-Since most evolutionary algorithms such as differential evolution were originally conceived to solve unconstrained problems, various constraint-handling techniques have been developed. One possible strategy is to generate and keep control variables in the feasible region as,

$$X_{ij}^{(G)} = \begin{cases} X_{ij}^{min} & , \text{if } X_{ij}^{(G)} \leq X_{ij}^{min} \\ X_{ij}^{max} & , \text{if } X_{ij}^{(G)} \geq X_{ij}^{max} \\ X_{ij}^{(G)} & , \text{otherwise} \end{cases}$$

i=1,.....N_P and j=1,.....D

where $X_{i,j}^{min}$ and $X_{i,j}^{max}$ are the lower and upper bounds of the jth decision parameter, respectively.

Penalty functions can be used whenever there are violations to some equality and/or inequality constraints [7]. Basically, the objective function f(x) is substituted by a fitness function f'(x) that penalizes the fitness whenever the solution contains parameters that violate the problem constraints,

$$f'(x) = f(x) + \text{Penalty}(x)$$

In this paper, the exterior penalty function method is applied to the equality constraints [7]. The new objective function is then given by

$$f'(X) = f(X) + \sum_{i=1}^{N_p} K_i [g_i(X)]^2$$

where K_i is a positive constant number, reflecting the constraint weight. The specification of these weighting factors depends on how strongly we feel about satisfying the constraints.

Application of DE to OPF:-Differential Evolution has been applied to problems from several areas. Some power engineering problems have been solved with DE including: Distribution systems capacitors placement, harmonics voltage

distribution reduction and passive shunt harmonic filter planning. DE has also been used in the design of filters, neural network learning, fuzzy logic application, and optimal control problems, among others.

The objective function of OPF,

$$F_{cost} = \sum_{i=1}^{n_g} F_i = \sum (a_i p_{gi}^2 + b_i p_{gi} + c_i) \text{ \$/Hr}$$

Subjected to constraints-

$$g(x, u) = 0$$

$$h(x, u) \leq 0$$

where g is the equality constraints and represent typical load flow equations. h is the system operating constraints.

Dependent Variables: X is the vector of dependent variables consisting of slack bus power P_{G1} load bus voltages V_L , generator reactive power outputs Q_G , and transmission line loadings S_l .

Hence, X can be expressed as

$$X^T = [P_{G1}, V_L, Q_G, S_l]$$

$$X^T = [P_{G1}, V_{L1}, \dots, V_{LN_{pq}}, Q_{G1}, \dots, Q_{GN_g}, S_{11}, \dots, S_{1N_l}]$$

Where N_{pq} , N_g , N_l are number of load buses, number of generators, and number of transmission lines respectively.

Independent Variables: U is the vector of independent variables consisting of generator voltages V_G , generator real power outputs P_G , except at the slack bus P_{G1} , and transformer tap settings T.

Hence, U can be expressed as

$$U = [V_G, P_G, T]$$

$$U^T = [V_{G1}, \dots, V_{GN_g}, P_{G2}, \dots, P_{GN_g}, T_1, \dots, T_{N_t}]$$

Where N_t is the number of the regulating transformers.

Initialization: The first step in this algorithm is to create an initial population. All the independent variables $[V_G, P_G, T]$ have to be generated according to formula (3), where each independent parameter of each individual in the population is assigned a value inside the given feasible region of the generator. This creates parent vectors of independent variables for the first generation. As they have created within their limits, they readily satisfy the corresponding inequality constraints. To find dependent variables corresponding to each individual, Newton-Raphson power flow solution is implemented. Fitness includes fuel cost function

and also penalties corresponding to dependent variables. Inclusion of these penalties in fitness gives us a great opportunity to assign better fitness to that particular population member whose control parameters are within the operational limits in addition to minimum fuel cost.

Where

$$F_{itp} = \frac{1}{A + (K_2 * \sum_{i=1}^{N_p} Q_{gp} f_{p,i}) + (K_3 * \sum_{i=1}^{N_p} V_p f_{p,i}) + (K_4 * \sum_{i=1}^{N_p} LF_p F_i)}$$

Where $A = F_{cost} + (K_1 * S_{pf})$

Slack bus penalty: Spf

Line flows penalty: Lfpf

QG Penalty : Qgpf

Voltage penalty : Vpf.

DE Algorithm Solution:- This optimization process is carried out with three basic operations:

- Mutation
- Cross over
- Selection

First, the mutation operation creates mutant vectors by perturbing each target vector with the weighted difference of the two other individuals selected randomly. Then, the cross over operation generates trail vectors by mixing the parameters of the mutant vectors with the target vectors, according to a selected probability distribution. Finally, the selection operator forms the next generation population by selecting between the trial vector and the corresponding target vectors those that fit better the objective function.

Initialization: The first step in the DE optimization process is to create an initial population of candidate solutions by assigning random values to each decision parameter of each individual of the population. Such values must lie inside the feasible bounds of the decision variable and can be generated by Eq. (3). In case a preliminary solution is available, adding normally distributed random deviations to the nominal solution often generates the initial population. for each value of j.

$$Y_{i,j}^{(0)} = Y_j^{min} + r_j (Y_j^{max} - Y_j^{min})$$

i= 1.....N_p,

j=1,2.....D

Where Y_j^{min} and Y_j^{max} are respectively, the lower and upper bound of the jth decision parameter and r_j is a uniformly distributed random number within

[1, 1] generated a new **Mutation:**

After the population is initialized, this evolves through the operators of mutation, cross over and selection. For crossover and mutation different types of strategies are in use. Basic scheme is explained here elaborately. The mutation operator

is in charge of introducing new parameters into the population. To achieve this, the mutation operator creates mutant vectors by perturbing a randomly selected vector (Y_a) with the difference of two other randomly selected vectors (Y_b and Y_c) according Eq. (4). All of these vectors must be different from each other, requiring the population to be of at least four individuals to satisfy this condition. To control the perturbation and improve convergence, the difference vector is scaled by a user defined constant in the range [0, 1.2]. This constant is commonly known as the scaling constant (S).

$$Y_i^{(G)} = Y_a^{(G)} + S (Y_b^{(G)} - Y_c^{(G)})$$

Where Y_a, Y_b, Y_c are randomly chosen vectors $\epsilon\{1,2,\dots,N_p\}$ and $a \neq b \neq c \neq i$

Y_a, Y_b, Y_c are generated anew for each parent , S is the scaling constant For certain problems it is considered a = i

Crossover:-The crossover operator creates the trial vectors, which are used in the selection process. A trail vector is a combination of a mutant vector and a parent (target) vector based on different distributions like uniform distribution, binomial distribution, exponential distribution is generated in the range [0, 1] and compared against a user defined constant referred to as the crossover constant. If the value of the random number is less or equal than the value of the crossover constant, the parameter will come from the mutant vector, otherwise the parameter comes from the parent vector.

$$X_{i,j}^{(G)} = \begin{cases} X_{i,j}^{(G)} & \text{if } r_j \leq CR \text{ or } j = q \\ X_{i,j}^{(G)} & \text{otherwise} \end{cases}$$

Where $i=1,2,\dots,N_p, j=1,2,\dots,D$

q is a randomly chosen index $\epsilon\{1,2,\dots,D\}$ that guarantees that the trial vector gets at least one parameter from the mutant vector. r_j is a uniformly distributed random number within [1,1] generated a new for each value of j . $X_{i,j}^{(G)}$

is the parent (target) vector, $X_{i,j}^{(G)}$ the mutant vector and $X_{i,j}^{(G)}$ the trial vector.

Selection: The selection operator chooses the vectors that are going to compose the population in the next generation. This operator compares the fitness of the trial vector and fitness of the corresponding target vector, and selects the one that performs better as mentioned in Eq.

$$Y_i^{(G+1)} = \begin{cases} Y_i^{(G)} & \text{if } f(Y_i^{(G)}) \leq f(Y_i^{(G)}) \\ Y_i^{(G)} & \text{otherwise} \end{cases}$$

i=1,2,.....N_p

II. RESULT AND DISCUSSION

All generator active power, and generator bus voltages and transformer tap setting are considered as continuous for simplicity. The generators cost coefficients of the IEEE 30- bus test system are given in the Table 2. Where 50 MW and 200 MW are the minimum and maximum real power output limits for bus no. 1 for IEEE 30 bus system and similarly 12 MW and 40 MW are the minimum and maximum real power output limits for bus no. 13 for IEEE 30 bus system. Cost coefficient c is zero throughout the process but value of cost coefficient b and c is given in the table corresponding to bus no.

TABLE II
 GENERATOR COST COEFFICIENTS OF IEEE 30-BUS SYSTEM

Bus No	Real Power Output limit (MW)		Cost Coefficients		
	Min	Max	a	b	c
1	50	200	0	2.00	0.00375
2	20	80	0	1.75	0.01750
5	15	50	0	1.00	0.06250
8	10	35	0	3.25	0.00834
11	10	30	0	3.00	0.02500
13	12	40	0	3.00	0.02500

DE Results and comparison with GA:-In this section, the DE solution of the OPF is evaluated using the test system IEEE-30 bus system .The results, which follow, are the best solution over the twenty runs. The results are compared with GA and GPM, SLP and QN.

DE Results compared with GA

Table IV

Parameters	SLP [17]	QN [08]	GA [08]	DE
Pg1(MW)	175.25	170.24	179.37	177.09
Pg2(MW)	48.34	44.95	44.24	48.92
Pg3(MW)	21.21	28.90	24.61	21.51
Pg4(MW)	23.60	17.48	19.90	21.27
Pg5(MW)	12.25	12.17	10.71	12.00
Pg6(MW)	12.33	18.47	14.09	12.00
Total	292.98	292.21	292.92	292.75
Loss(MW)	9.57	8.80	9.52	9.40
Cost(\$/h)	803.08	807.78	803.69	801.84
CPUtime	1.12	N/A	7.00	5.98

Result Analysis: Table 4 shows the comparison of the cost, Losses and time of generation for the IEEE-30 bus system for the above cases with other soft computing methods. From the table it can be shown very clearly for same load demand, generating cost and losses are different. Results

obtained by DE are much better then other evolutionary techniques DE converges very fast give better results in lesser time. generating cost obtained by DE is 801.84 \$/h where by GA is 803.69. Where time taken by DE is 5.98 sec and time taken by GA is 7 sec. cost obtained by other methods as GPM is 804.85\$/h , SLP is 803.08 \$/h and by QN is 807.78. so results obtained by other methods are not satisfactory.

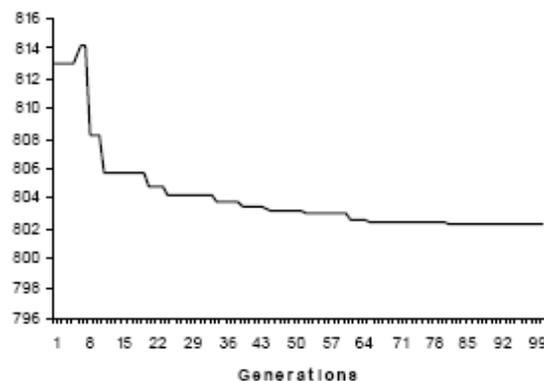


Figure 5 shows the convergence of DE for the optimal power flow problem. The operating costs of the best solution in the normal operation achieved by the DE and GA are, respectively, \$801.84 and \$803.78 per hour. It can be observed from Fig.3 that the convergence of DE is faster while obtaining a better solution in lesser computational time. GA converges in 7 sec for twenty generations while DE converges in 5.98 sec for same generations.

III. CONCLUSION

The work presents a DE solution to the optimal power flow problem and is applied to an IEEE 30-bus power system. The main advantage of DE over other modern heuristics is modeling flexibility, sure and fast convergence, less computational time than other heuristic methods. And it can be easily coded to work on parallel computers. The main disadvantage of DE is that it is heuristic algorithms, and it does not provide the guarantee of optimal solution for the OPF problem. The DE approach is useful for obtaining high quality solution in a very less time compared to other methods.

The future work in this area consists of the applicability of DE solutions to large-scale of problems of systems with several thousands of nodes, utilizing the strength of parallel computers.

REFERENCES

- [1]. H. W. Dommel, W. F. Tinney, Optimal Power Flow Solutions, IEEE Transactions

- on power apparatus and systems, Vol. PAS-87, No. 10, p. 1866-1876, October 1968..
- [2]. K. Y. Lee, Y.M. Park, and J.L. Ortiz, A United Approach to Optimal Real and Reactive Power Dispatch, IEEE Transactions on Power Systems, Vol. PAS-104, p.1147-1153, May 1985.
- [3]. M. Sasson, Non linear Programming Solutions for load flow, minimum loss, and economic dispatching problems, IEEE Trans. on power apparatus and systems, Vol. PAS-88, No. 4, April 1969.
- [4]. T. Bouktir, M. Belkacemi, K. Zehar, Optimal power flow using modified gradient method, Proceeding ICEL'2000, U.S.T.Oran, Algeria, Vol. 2, p. 436-442, 13-15 November 2000.
- [5]. R. Fletcher, Practical Methods of Optimisation, John Willey & Sons, 1986
- [6]. R.Gnanadas, P.Venkatesh, Narayana Prasad Padhy, "Evolutionary Programming Based Optimal Power Flow For Units With Non-Smooth Fuel Cost Functions", Electric Power Components and Systems, Vol.33, 2005, pp. 1245-1250.
- [7]. Jason Yuryevich, Kit Po Wong, "Evolutionary Programming Based Optimal Power Flow Algorithm", IEEE Transactions on Power Systems, Vol. 14, No. 4, November 1999, pp.1245-1250.
- [8]. Tarek Bouktir, Linda Slimani, "Genetic Algorithm to solve the Optimal power flow problem". Leonardo journal of science January- June 2004.
- [9]. Wong K. P., Yuryevich J., Optimal power flow method using evolutionary programming, Springer-Verlag Berlin, 1999, p. 405-412.

Tarabhan Gupta "Optimal Power Flow Using Differential Evolution" "International Journal of Engineering Research and Applications (IJERA) , vol. 7, no. 12, 2018, pp. 61-67