ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

Hybrid Differential Evolution and Particle Swarm Optimization Based Solutions for Economic Load Dispatch Problem

A. Evangelin Lara[1], A. Josephine Amala[2]

PG Student [PSE], Dept. of EEE, JJ College of Engineering, Tiruchirappalli, Tamilnadu, India[1]

Professor, Dept. of EEE, JJ College of Engineering, Tiruchirappalli, Tamilnadu, India[2]

ABSTRACT: In this paper, a hybrid differential evolution and a particle swarm optimization based algorithms are proposed for solving the problem of scheduling the economic dispatch problems. The efficient scheduling requires minimizing the operating cost of the power system. To improve the global optimization property of DE, the PSO procedure is integrated as additional mutation operator. The effectiveness of the proposed algorithm (termed DEPSO) is demonstrated by solving ELD problems with nonsmooth and nonconvex solution spaces. The comparative results with some of the most recently published methods confirm the effectiveness of the proposed strategy to find accurate and feasible optimal solutions for practical ELD problems. The proposed DEPSO algorithms are implemented for a test system. The program has been developed in Matlab platform. It is shown that the proposed techniques yield optimal solutions when compared to other non-conventional methods.

KEYWORDS: Economic load dispatch, Differential evolution, Particle swarm optimization, Optimization, hybrid DEPSO,

I.INTRODUCTION

Economic load dispatch is a fundamental function in modern power system operation and control. The economic load dispatch (ELD) problem of power generation involves allocation of power generation to different thermal units to minimize the total fuel cost while satisfying load demand and diverse operating constraints of a power system [1]. The traditional ELD considers the power balance constraint apart from the generating capacity limits. However, due to practical limitations in operation and control of power systems, realistic ELD must take a variety of practical constraints into consideration to provide the completeness for the ELD problem formulation.

II. RELATED WORKS

In practice, real input–output characteristics present higher order nonlinearities and discontinuities due to valve-point loading effects caused by the sharp increase in losses when steam admission valves are first opened [2]. However, generating units may have prohibited operating zones due to faults in the machines themselves or the associated auxiliaries, such as boilers, feed pumps, leading to instabilities in certain ranges of the unit loading [3]. In addition, many generating units need the cost function to be modelled as piecewise function, due to their capability of operating with multi-fuel sources (coal, nature gas, or oil), leading to the problem of determining the most economic fuel to burn [4]. Furthermore, due to the fact that unit generation output cannot be changed instantaneously, the unit in the actual operating processes is restricted by its ramp rate limits [5]. Also, for security and reliability considerations of power systems, spinning reserve capacity must be sufficient to absorb source contingencies and major load forecast errors without load shedding [1, 6].

The above operation constraints and nonlinearities make the ELD problem a nonsmooth optimization problem having complex and nonconvex features with heavy equality and inequality constraints. Conventional gradient based optimization methods are not capable to solve efficiently this kind of problems and usually result in inaccurate

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

dispatches causing huge loss of revenue over the time. Dynamic programming [1] can treat such types of problems, but this method is computationally extensive, and suffers from the problem of dimensionality.

Differential Evolution (DE) and Particle Swarm Optimization (PSO) algorithms are two of these recently developed heuristic global search tools. The DE algorithm has been introduced by Storn and Price in 1995 [7]. DE algorithm is a stochastic population based search method successfully applied in global optimization problems. DE improves a population of candidate solutions over several generations using the mutation, crossover and selection operators in order to reach an optimal solution [8]. DE presents great convergence characteristics and requires few control parameters, which remain fixed throughout the optimization process and need minimum tuning. PSO refers to a relatively new family of algorithms that may be used to find optimal or near optimal solutions to numerical and qualitative problems. PSO was developed by Kennedy and Eberhartin 1995 [9] and inspired by social behaviour of bird flocking and fish schooling. PSO has proven to be both very fast and effective when applied to a diverse set of optimization problems

III. OVERVIEW OF DIFFERENTIAL EVOLUTION ALGORITHM

The differential evolution algorithm (DE) is a population based algorithm like genetic algorithm using the similar operators; crossover, mutation and selection. In DE, each decision variable is represented in the chromosome by a real number. As in any other evolutionary algorithm, the initial population of DE is randomly generated, and then evaluated. After that, the selection process takes place. During the selection stage, three parents are chosen and they generate a single offspring which competes with a parent to determine which one passes to the following generation. DE generates a single offspring (instead of two like in the genetic algorithm) by adding the weighted difference vector between two parents to a third parent. If the resulting vector yields a lower objective function value than a predetermined population member, the newly generated vector replaces the vector to which it was compared.

An optimization task consisting of D parameters can be presented by a D-dimensional vector. In DE, a population of NP solution vectors is randomly created at the start. This population is successfully improved over G generations by applying mutation, crossover and selection operators to reach an optimal solution [7, 8]. The main steps of the DE algorithm are given below:

- Initialization
- Evaluation
- Repeat
- Mutation
- Crossover
- Evaluation
- Selection Until (Termination criteria are met)

The termination criteria are the conditions under which the search process will stop. In this work, the search procedure will terminate whenever the predetermined maximum number of generations G^{max} is reached, or whenever the global best solution does not improve over a predetermined number of generations.

A. Initialization

Typically, each decision parameter in every vector of the initial population is assigned a randomly chosen value from within its corresponding feasible bounds:

$$x_{j,i}^{(0)} = x_j^{mix} + \mu_j \,(x_j^{max} - x_j^{mix}),\; i=1,\ldots, NP,\; j=1,\ldots, D \qquad -(1)$$

B. Mutation

The mutation operator creates mutant vectors x_i by perturbing a randomly selected vector x_a with the difference of two other randomly selected vectors x_b and x_c, according to the following equation

$$x'^{(G)}_i = x_a^{(G)} + \alpha(x_b^{(G)} - x_c^{(G)}),\; i=1,\ldots, NP \qquad -(2)$$

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

C. Crossover

The crossover operation generates trial vectors xi by mixing the parameters of the mutant vectors xi with its target or parent vectors xi, based on a series of D−1 binomial experiments of the following form

$$x''^{(G)}_{j,i} = \begin{cases} x'^{(G)}_{j,i} \text{ if } \rho_j \leq C_R \text{ or } j = q, \\ x^{(G)}_{j,i}, \text{ otherwise} \quad, i=1,\ldots, NP, j=1, D \end{cases} \qquad -(3)$$

D. Selection

The selection operator forms the population by choosing between the trial vectors and their predecessors (parent vectors) those individuals that present a better fitness or are more optimal according to

$$x^{(G+1)}_{i} = \begin{cases} x''^{(G)}_{i} \text{ if } f(x''^{(G)}_{i}) \leq f(x^{(G)}_{i}), \\ x^{(G)}_{i}, \text{ otherwise} \quad, i=1,\ldots, NP \end{cases} \qquad -(4)$$

IV. OVERVIEW OF PARTICLE SWARM OPTIMIZATION

The particle swarm optimization (PSO) is a swarm intelligence method that differs from well-known evolutionary computation algorithms, such as genetic algorithms (GA), in that the population is not manipulated through operators inspired by the human DNA procedures. Instead, in PSO, the population dynamics simulate the behaviour of a "birds' flock", where social sharing of information takes place and individuals profit from the discoveries and previous experience of all other companions during the search for food. Thus, each companion, called 'particle', in the population, which is called 'swarm', is assumed to "fly" over the search space looking for promising regions on the landscape. For a minimization case, such regions possess lower function values than others previously visited. In this context, each particle is treated as a point into the search space, which adjusts its own flying according to its flying experience as well as the flying experience of other particles. Therefore, each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) that it has achieved so far. This implies that each particle has a memory, which allows it to remember the best position on the feasible search space that it has ever visited. This value is commonly called Pbest. Another best value that is tracked by the particle swarm optimizer is the best value obtained so far by any particle in the neighbourhood of the particle. This location is commonly called Gbest. The basic idea behind the particle swarm optimization technique consists, at each iteration, updating the velocity and accelerating each particle towards Pbest and Gbest locations [9].

The velocity of each particle can be modified by using the following equation:

$$V_i^{k+1} = \omega\, V_i^k + c_1 r_1 (Pbest_i^k - X_i^k) + c_2 r_2 (Gbest^k - X_i^k) \qquad -(5)$$

c2 are positive constants, and r1, r2 are random numbers within the range [0,1]. The position of each particle is updated by the following equation:

$$X_i^{k+1} = X_i^k + V_i^{k+1} \qquad -(6)$$

The positive constants c1, c2 provide the correct balance between exploration and exploitation, and are called the cognitive parameter and the social parameter, respectively. The random numbers provide a stochastic characteristic for the particles velocities in order to simulate the real behaviour of the birds in a flock. The weight parameter ω is a control parameter which is used to control the impact of the previous history of velocities on the current velocity of each particle. Hence, the parameter ω regulates the trade-off between global and local exploration ability of the swarm. The recommended value of the inertia weight ω is to set it to a large value for the initial stages, in order to enhance the

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

global exploration of the search space, and gradually decrease it to get more refined solutions facilitating the local exploration in the last stages. In general, the inertia weight factor is set according to the following equation:

$$\omega = \omega_{\max} - \frac{\omega\, max - \omega min}{iter\ max} iter \qquad - (7)$$

The velocity of each particle is limited by a maximum value V^{max}_i which facilitates local exploration of the problem space and it realistically simulates the incremental changes of human learning. This limit is given by

$$V_i^{max} = (X_i^{max} - X_i^{min}) / N \qquad - (8)$$

V. DIFFERENTIAL EVOLUTION AND PARTICLE SWARM OPTIMIZATION HYBRIDS

As discussed above, DE algorithm has some advantages, such as its ability to maintain the diversity of population and to explore local search, but it does not have a mechanism to memorize the previous process and uses the global information about the search space, so it easily results in a waste of computing power leading to premature convergence. On the other hand, PSO may easily get trapped in local minima because of swarm diversity loss.

Inspired by their advantages and their disadvantages, we propose in this paper a new hybrid optimization technique, namely DEPSO, by combining differential evolution algorithm and particle swarm optimization. This new hybrid optimization approach has been proposed by Hao et al. [10]. The basic idea is to incorporate the PSO procedure as a supplementary mutation operator into the conventional DE algorithm to improve the global search capability, and thus avoiding the premature convergence to local minima. The proposed DEPSO consists essentially in a strong co-operation of the two evolutionary algorithms described above, since it maintains the integration of the two techniques for the entire run. The general structure of the proposed DEPSO algorithm is similar to that of DE. The main difference is in the mutation operation. Instead of manipulating a single mutant vector in DE, we use two mutant vectors in DEPSO. The first is the classical mutant vector of DE, and the second is generated using the PSO updating rules for particle position [10]. Therefore, in the crossover operation, two different trial vectors are created for each parent or target vector. The selection operator form a new individual by choosing among the trial vectors and the parent vector, i.e. the individual that presents a better fitness or are more optimal. This efficient combination strategy of DE and PSO improves the global search capability, avoiding convergence to local minima and at the same time accelerates the convergence.

Solution methodology

The proposed DEPSO algorithm applied in this study for ELD problem can be described as follows:

Step 1: Initialize the vectors of candidate solutions of the parent population (particles), between the maximum and minimum operating limits of the generating units.
Step 2: Generate the particle velocities randomly.
Step 3: Evaluate the fitness function of each particle
Step 4: Obtain the first mutant vector using DE rule
Step 5: Compute the particle velocity.
Step 6: Obtain the second mutant vector using the PSO updating rule.
Step 7: Crossover of control variables to generate two trial vectors.
Step 8: Evaluation of new solution fitness.
Step 9: Selection operation.
Step 10: If one of the stopping criteria is met, than stop. Otherwise, go to step 4.

The stopping criteria are the conditions under which the search process will stop. In this work, the search procedure will terminate whenever the predetermined maximum number of generations G^{max} is reached, or whenever the global best solution does not improve over a predetermined number of generations.

A penalty function approach is used to handle the power balance constraint and inequality constraints [10]. The extended objective function FT (or fitness function) is defined by

$$F_T = F + k\left(\sum_{i=0}^{ng} P_i - P_D - P_L\right)^2 + \sum_{j=0}^{nc} PF_j \qquad - (9)$$

where K denotes the penalty factor of the equality constraint; nc represents the number of inequality constraints and PF_j is the penalty function for the jth inequality constraint; given as

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

$$P\,F_j = \begin{cases} K_j\,(U_j - U_j^{lim})^2 & \text{if violated} \\ 0 & \text{otherwise} \end{cases} \quad - (10)$$

K_j is the penalty factor for the j^{th} inequality constraint; U_j^{lim} is the limit value of the variable U_j.

VI. RESULTS AND DISCUSSION

In order to evaluate the performance of the proposed DEPSO approach, it was applied to four different kinds of ELD problems with nonsmooth and nonconvex solution spaces. The first case study considers the 40-unit system with valve-point effects [11]. The second one comprises the 15-unit system with quadratic cost curve, prohibited operating zones, ramp rate limits, and transmission network loss [12]. The third test case consists of the 10-unit system considering both valve-point effects and multiple fuels [13]. The fourth case study consists of the 10-unit system with valve point effects, multiple fuels, prohibited operating zones, ramp rate limits and spinning reserve [14]. All simulations were performed on a personal computer (i3 3.1 GHz Intel Processor and 2 GB RAM running MATLAB R2013a).

For the sake of conciseness, only the test system is used to examine the influence of DEPSO control parameter. To investigate how the parameters N_P, and C_R affect the performance of the proposed DEPSO, the following procedure has been applied:

1. The population size is fixed at 70 (chosen by trial and error).
2. Mutation factor ˛ and crossover constant CR are increased from 0.1 to 0.9 in suitable steps as shown in Table.

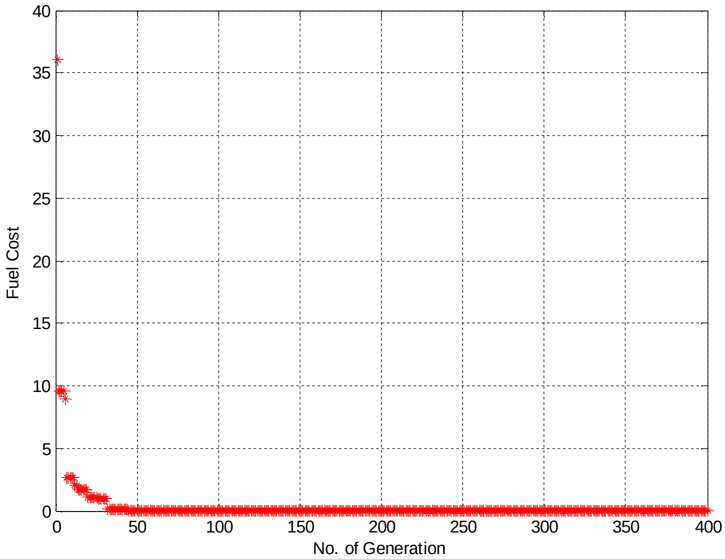

Fig.1 Optimum fuel costs obtained at different runs of DEPSO

For all above mentioned parameter combinations, performance of DEPSO is evaluated with 100 trial runs on the fourth test case. The results of this performance analysis are reported in Tables 0 which show the average fuel costs and best fuel costs under different parameter combinations. From Table-1, it is observed that DEPSO almost acquires the same average generation cost. These results reveal that the proposed hybrid approach is robust under various parameter combinations when applied to solve nonsmooth/nonconvex economic dispatch problems. The most noticeable observation from this groundwork is that the optimal settings for ˛ and CR are found to be 0.9 and 0.5, respectively, which give the optimal best cost of 624.354 $/h. Once optimal values of ˛ and CR have been obtained, the effect of population size is explored. The optimum population size is found to be related to the dimension and the complexity of

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

the problem. Results of 100 trial runs with varying population size of DEPSO are listed in Table. With increase in population, a steady improvement in average cost was observed. A population of 70 was found optimum for this test system.

Table: 1 Minimization of Fuel Cost

Test Result	FUEL COST ($/h)		
	Minimum Cost	Maximum Cost	Mean Cost
DEPSO	412.35	424.30	418.30
DE	424.48	425.80	424.60
PSO	424.54	425.60	424.56

The comparison of the minimum, maximum, mean cost and average computation time for the simulation over 100 trial runs obtained by the proposed hybrid DEPSO with the DE and PSO algorithms. As seen from Table 1, the best, worst and average solutions obtained with the proposed DEPSO algorithm are better than the best, worst and average results of all other methods over the four test systems. It is also observed that the worst solution of the proposed DEPSO algorithm outperforms all the best solutions found by the DE and DERAND approaches, Moreover, the average CPU time of the DEPSO is better or comparable with that of DE and DEPSO, which shows the reliability of the proposed algorithm.

VII.CONCLUSION

In this paper, a new efficient hybrid differential evolution based particle swarm optimization strategy was successfully implemented to solve practical nonsmooth and nonconvex economic dispatch problems. The PSO process was incorporated as a supplementary mutation operator into the conventional DE algorithm to improve the global search capability. In order to illustrate the application of the proposed method, it has been tested and examined with four test systems. The comparison of the results with those reported recently in the literature, confirms the superiority of the proposed method and its potential to find accurate and feasible optimal solutions for practical ELD problems, without any restrictions on the kind of the problems. In addition, the superior features of the algorithms are i) Simple and efficient. ii) Any number and types of constraints can be easily accommodated. iii) Suitable for solving any type of objective function (irrespective of the shape). iv) Reduced computing time. v) Smooth and fast convergence and vi) Better quality solutions. The evolutionary algorithms are still in the development stage and its implementation for online applications needs further research.

REFERENCES

[1] A.J. Wood, B.F. Wollenberg, "Power Generation, Operation and Control". 2nd ed., John Wiley & Sons, New York, 1996.
[2] D.C. Walters, G.B. Sheble, "Genetic algorithm solution of economic dispatch with valve point loading", IEEE Transactions on Power Systems 8 (3) (1993) 1325–1332.
[3] S.O. Orero, M.R. Irving, "Economic dispatch of generators with prohibited operating zones: a genetic algorithm approach", IEE Proceedings on Generation, Transmission & Distribution 143 (6) (1996) 529–534.
[4] C.L. Chiang, "Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels", IEEE Transactions on Power Systems 20 (4) (2005) 1690–1699.
[5] A. Pereira-Neto, C. Unsihuay, O.R. Saavedra, "Efficient evolutionary strategy optimization procedure to solve the nonconvex economic dispatch problem with generator constraints", IEE Proceedings on Generation, Transmission & Distribution 152 (5) (2005) 653–660.
[6] F.N. Lee, A.M. Breipohl, "Reserve constrained economic dispatch with prohibited operating zones", IEEE Transactions on Power Systems 8 (1) (1993) 246–254.
[7] R. Storn,K. Price, "Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces", Journal of Global Optimization 11 (4) (1997) 341–359.
[8] R. Storn, "On the usage of differential evolution for function optimization", in: Proceedings of Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS), Berkley, 1996, pp. 519–523.
[9] J. Kennedy, R.C. Eberhart, "Particle swarm optimization", in: Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, 1995, pp. 1942–1948.
[10] Z.F. Hao, G.H. Guo, H. Huang, "A particle swarm optimization algorithm with differential evolution", in: Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 2007, pp. 1031–1035.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

[11] N. Sinha, R. Chakrabarti, P.K. Chattopadhyay, “Evolutionary programming techniques for economic load dispatch”, IEEE Transactions on Evolutionary Computation 7 (1) (2003) 83–94.

[12] Z.L. Gaing, “Particle swarm optimization to solving the economic dispatch considering the generator constraints”, IEEE Transactions on Power Systems 18 (3) (2003) 1187–1195.

[13] C.L. Chiang, “Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels”, IEEE Transactions on Power Systems 20 (4) (2005) 1690–1699.

[14] N. Amjady, H. Nasiri-Rad, “Economic dispatch using an efficient real-coded genetic algorithm”, IET Generation, Transmission & Distribution 3 (3) (2009) 266–278.