World Academy of Science, Engineering and Technology
International Journal of Energy and Power Engineering
Vol:5, No:10, 2011

An Intelligent Water Drop Algorithm for Solving Economic Load Dispatch Problem

S. Rao Rayapudi

Abstract—Economic Load Dispatch (ELD) is a method of determining the most efficient, low-cost and reliable operation of a power system by dispatching available electricity generation resources to supply load on the system. The primary objective of economic dispatch is to minimize total cost of generation while honoring operational constraints of available generation resources. In this paper an intelligent water drop (IWD) algorithm has been proposed to solve ELD problem with an objective of minimizing the total cost of generation. Intelligent water drop algorithm is a swarm-based nature-inspired optimization algorithm, which has been inspired from natural rivers. A natural river often finds good paths among lots of possible paths in its ways from source to destination and finally find almost optimal path to their destination. These ideas are embedded into the proposed algorithm for solving economic load dispatch problem. The main advantage of the proposed technique is easy is implement and capable of finding feasible near global optimal solution with less computational effort. In order to illustrate the effectiveness of the proposed method, it has been tested on 6-unit and 20-unit test systems with incremental fuel cost functions taking into account the valve point-point loading effects. Numerical results shows that the proposed method has good convergence property and better in quality of solution than other algorithms reported in recent literature.

Keywords—Economic load dispatch, Transmission loss, Optimization, Valve point loading, Intelligent Water Drop Algorithm.

I. Introduction

THE operating cost of a power plant mainly depends on the fuel cost of generators and is minimized via economic load dispatch. Economic load dispatch problem can be defined as determining the least cost power generation schedule from a set of on line generating units to meet the total power demand at a given point of time [1]. The main objective of ELD problem is to decrease fuel cost of generators, while satisfying equality and inequality constraints. In this problem, fuel cost of generation is represented as cost curves and overall calculation minimizes the operating cost by finding a point where total output of generators equals total power that must be delivered plus losses.

In conventional economic load dispatch, cost function for each generator has been approximately represented by a single quadratic function and is solved using lambda iteration method, gradient-based method, etc. [2]. These methods require incremental fuel cost curves which are piecewise linear and monotonically increasing to find the global optimal solution. For generating units, which actually having non-monotonically incremental cost curves, conventional methods ignores or flattens out portions of incremental cost curve that are not continuous or monotonically increasing. Unfortunately, input-output characteristics of modern units are inherently highly non-linear because of valve point loadings, ramp rate limits, prohibiting operating zones etc., resulting in multiple local minimum points in the cost function. So, their characteristics have to be approximated to meet requirements of classical dispatch algorithms. However, such approximations may lead to huge loss of revenue over the time. Consideration of highly nonlinear characteristics of units demand for solution techniques having no restrictions on shape of fuel cost curves [3]- [4].

Classical methods like Newton-based and gradient methods cannot perform very well for problems having highly nonlinear characteristics with large number of constraints and many local optimum solutions. Dynamic programming is one of the approache to solve non-linear and discontinuous ELD problem, but it suffers from problem of curse of dimensionality or local optimality [5]. Methods based on artificial intelligence techniques, such as artificial neural networks, are presented [6]- [9]. However, neural network-based approaches may suffer from excessive numerical iterations, resulting in huge calculations. Heuristic search techniques, such as particle swarm optimization [10], genetic algorithms [11]- [13], differential evolution [14] and tabu search [15] have also been successfully applied to ELD problems. Recently, biogeography-based optimization [16] is proposed to solve the ELD problem.

In this paper, a new approach is proposed to solve non-smooth ELD problem with valve-point effect using intelligent water drop (IWD) algorithm [17]. intelligent water drop algorithm imitate some of the processes that happen in nature between the water drops of a river and the soil of the river bed. The performance, effectiveness, and robustness of the proposed method are assessed via intensive testing and comparison of results with other methods reported in recent literature.

The rest of this paper is organized as follows. Next section of the paper presents formulation of ELD problem as a constrained optimization problem. Section III describes intelligent water drop algorithm. IV reports test results. The paper ends in Section V with a brief discussion on results.

II. Formulation of Optimization Problem

The objective of economic load dispatch problem is to find optimal combination of power generations that minimizes total cost generation while satisfying different equality and inequality constraints. Thus, the optimization problem is formulated as follows.

Dr. S. Rao Rayapudi is with the Department of Electrical and Electronics Engineering, J. N. T. University Kakinada, INDIA, e-mail: sraor@in.com

Consider a power plant having N generating units, each loaded to P_i MW. The units should be loaded in such a way that total fuel cost C_T of generators should be minimum while satisfying the operating constraints. Therefore, objective function of classical ELD problem is written as:

$$Minimize \quad C_T = Min. \quad \sum_{i=1}^{N} C_i(P_i) \tag{1}$$

where C_i is the fuel inputpower output cost function of i^{th} generator. Usually, fuel cost of the generating thermal unit is expressed as a second order approximate function of its output P_i.

$$C_i(P_i) = a_i + b_i P_i + c_i P_i^2 \tag{2}$$

where a_i, b_i and c_i are fuel cost coefficients of unit *i*. To take care of valve-point effects, a sinusoidal function are added to the fuel cost function and is represented as:

$$\tilde{C}_i(P_i) = a_i + b_i P_i + c_i P_i^2 + |e_i \sin(f_i(P_i^{min} - P_i))| \tag{3}$$

where e_i and f_i are the coefficients of generator *i* reflecting the valve-point effects.

By including power balance, generation limits, and prohibited operating zone constraints, overall objective function is written as

$$\begin{aligned} Min. \quad \tilde{C}_T = & \sum_{i=1}^{N} \tilde{C}_i(P_i) \\ & + \lambda_1 \sum_{i=1}^{N} \{P_i - P_D - P_{Loss}\} + \lambda_2 \sum_{j=1}^{N} V_j^k \end{aligned} \tag{4}$$

where λ_1 and λ_2 are positive constants (penalty factors) associated with the power balance and prohibited zones constraints, respectively. For ELD problem without transmission loss and prohibited zone constraints, the setting $\lambda_1 = \lambda_2 = 0$ is most rational. However, for ELD problems with transmission loss and prohibited zone constraints these factors are tuned empirically and their values were set as $\lambda_1 = 1$ and $\lambda_2 = 5 \times N$ in all cases. V_j^k reckons the violation of prohibited zone constraints for the individual *j*, can be defined as

$$V_j^k = \begin{cases} 1, & if \quad P_j \quad voilates \quad prohibited \quad zone \\ 0, & otherwise \end{cases} \tag{5}$$

The expressions for power balance, generation limits, and prohibited operating zones constraints are given as follows.

1) Power balance constraint:

$$\sum_{i=1}^{N} P_i = P_D + P_{Loss} \tag{6}$$

where P_D is total load demand and P_{Loss} is total transmission loss, which depends on physical geographical network and P_i is generated power level from each unit. Calculation of P_{Loss} using B-matrix loss coefficients is expressed as

$$P_{Loss} = \sum_{i=1}^{N}\sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1}^{N} B_{i0} P_i + B_{00} \tag{7}$$

where B_{ij} is ij^{th} element of loss coefficient symmetric matrix B, B_{i0} is i^{th} element of loss coefficient vector and B_{00} is loss coefficient constant.

2) Generation limits: The generating capacity constraint is given by

$$P_i^{min} \le P_i \le P_i^{max} \tag{8}$$

where P_i^{min} and P_i^{max} are minimum and maximum power outputs of i^{th} unit. This constraint prohibits a cheap unit to generate power more than its maximum limit as well as an expensive unit to generate power less than its minimum limit.

3) Prohibited operating zones constraint: In practical operations, generated output P_i of unit *i* must avoid operations in prohibited zones. The feasible operating zones of unit *i* can be described as

$$P_i \in \begin{cases} P_i^{min} \le P_i \le P_{i,1}^l \\ P_{i,k-1}^u \le P_i \le P_{i,k}^l & (k = 2, 3, ...X_i) \\ P_{i,X_i}^u \le P_i \le P_i^{max} \end{cases} \tag{9}$$

where $P_{i,k}^l$ and $P_{i,k}^u$ are lower and upper bounds of the k^{th} prohibited zone of unit *i*, and X_i is number of prohibited zones of unit *i*.

III. INTELLIGENT WATER DROPS ALGORITHM

A. Overview of Intelligent Water Drops Algorithm

Intelligent Water Drops algorithm (IWD) [18] is a swarm-based nature-inspired optimization algorithm, which has been inspired from natural rivers and how they find almost optimal path to their destination. A natural river often finds good paths among lots of possible paths in its ways from the source to destination. These near optimal or optimal paths follow from actions and reactions occurring among the water drops and the water drops with their riverbeds. In the IWD algorithm, several artificial water drops cooperate to change their environment in such a way that the optimal path is revealed as the one with the lowest soil on its links. The solutions are incrementally constructed by the IWD algorithm. Consequently, the IWD algorithm is generally a constructive population-based optimization algorithm. The Intelligent Water Drop, IWD for short, flows in its environment has two important properties:

1. The amount of the soil it carries now, Soil (IWD).
2. The velocity that it is moving now, Velocity (IWD).

This environment depends on the problem at hand. In an environment, there are usually lots of paths from a given source to a desired destination, which the position of the destination may be known or unknown. If we know the position of the destination, the goal is to find the best (often the shortest) path from the source to the destination. In some cases, in which the destination is unknown, the goal is to find the optimum destination in terms of cost or any suitable measure for the problem.

We consider an IWD moving in discrete finite-length steps. From its current location to its next location, the IWD velocity is increased by the amount nonlinearly proportional to the inverse of the soil between the two locations. Moreover, the IWDs soil is increased by removing some soil of the path joining the two locations. The amount of soil added to the IWD is inversely (and nonlinearly) proportional to the time needed for the IWD to pass from its current location to the next location. This duration of time is calculated by the simple laws of physics for linear motion. Thus, the time taken is proportional to the velocity of the IWD and inversely proportional to the distance between the two locations. Another mechanism that exists in the behavior of an IWD is that it prefers the paths with low soils on its beds to the paths with higher soils on its beds. To implement this behavior of path choosing, we use a uniform random distribution among the soils of the available paths such that the probability of the next path to choose is inversely proportional to the soils of the available paths. The lower the soil of the path, the more chance it has for being selected by the IWD.

B. Intelligent Water Drops Algorithm

The IWD algorithm gets a representation of the problem in the form of a graph (N, E) with the node set N and edge set E. Then, each IWD begins constructing its solution gradually by traveling on the nodes of the graph along the edges of the graph until the IWD finally completes its solution. One iteration of the algorithm is complete when all IWDs have completed their solutions. After each iteration, the iteration-best solution T^{IB} is found and it is used to update the total-best solution T^{TB}. The amount of soil on the edges of the iteration-best solution T^{IB} is reduced based on the goodness (quality) of the solution. Then, the algorithm begins another iteration with new IWDs but with the same soils on the paths of the graph and the whole process is repeated. The algorithm stops when it reaches the maximum number of iterations $iter_{max}$ or the total-best solution T^{TB} reaches the expected quality. The IWD algorithm has two kinds of parameters. One kind is those that remain constant during the lifetime of the algorithm and they are called 'static parameters'. The other kind is those parameters of the algorithm, which are dynamic and they are reinitialized after each iteration of the algorithm.

The algorithm of IWD is specified in the following steps:

1. The graph (N, E) of the problem is given to the algorithm. The quality of the total-best solution T^{TB} is initially set to the worst value: $q(T^{TB}) = \infty$. The maximum number of iterations $iter_{max}$ is specified by the user. The iteration count $iter_{count}$ is set to zero. The number of water drops N_{IWD} is set to a positive integer value, which is usually set to the number of nodes N_c of the graph. For velocity updating, the parameters are a_v =1, b_v =0.01 and c_v = 1. For soil updating, a_s =1, b_s =0.01 and c_s = 1. The local soil updating parameter ρ_n=0.9, which is a small positive number less than one. The global soil updating parameter ρ_{IWD}=0.9, which is chosen from [0, 1]. Moreover, the initial soil on each path (edge) is denoted by the constant *InitSoil* such that the soil of the path between every two nodes *i* and *j* is set by $soil(i, j) = InitSoil$. The initial velocity of each IWD is set to $InitVel$. Both parameters $InitSoil$ and $InitVel$ are user selected and they should be tuned experimentally for the application.
2. Every IWD has a visited node list $V_c(IWD)$, which is initially empty: $V_c(IWD)$= . Each IWDs velocity is set to $InitVel$. All IWDs are set to have zero amount of soil.
3. Spread the IWDs randomly on the nodes of the graph as their first visited nodes.
4. Update the visited node list of each IWD to include the nodes just visited.
5. Repeat Steps 5.1 to 5.4 for those IWDs with partial solutions.

5.1 For the IWD residing in node *i*, choose the next node *j*, which does not violate any constraints of the problem and is not in the visited node list $V_c(IWD)$ of the IWD, using the following probability $p_i^{IWD}(j)$:

$$p_i^{IWD}(j) = \frac{f(soil(i,j))}{\sum_{k \notin V_c(IWD)} f(soil(i,k))} \tag{10}$$

such that

$$f(soil(i,j)) = \frac{1}{\epsilon_s + g(soil(i,j))}$$

and

$$g(soil(i,j)) = \begin{cases} soil(i,j) & if \min_{l \notin V_c(IWD)} (soil(i,l)) \geq 0 \\ soil(i,j) - \min_{l \notin V_c(IWD)} (soil(i,l)) & else \end{cases}$$

Then, add the newly visited node *j* to the list $V_c(IWD)$.

5.2 For each IWD moving from node *i* to node *j*, update its velocity $vel^{IWD}(t)$ by

$$vel^{IWD}(t+1) = vel^{IWD}(t) + \frac{a_v}{b_v + c_v.soil^2(i,j)} \tag{11}$$

where $vel^{IWD}(t+1)$ is the updated velocity of the IWD.

5.3 For the IWD moving on the path from node *i* to node *j*, compute the soil $\Delta soil(i,j)$ that the IWD loads from the path by

$$\Delta soil(i,j) = \frac{a_s}{b_s + c_s.time^2(i,j;vel^{IWD}(t+1))} \tag{12}$$

such that

$$time(i,j;vel^{IWD}(t+1)) = \frac{HUD(j)}{vel^{IWD}(t+1)}$$

where the heuristic undesirability $HUD(j)$ is defined appropriately for the given problem.

5.4 Update the soil $soil(i,j)$ of the path from node *i* to node *j* traversed by that IWD and also update the soil that the IWD carries $soil^{IWD}$ by

$$soil(i,j) = (1-\rho_n).soil(i,j) - \rho_n.\Delta soil(i,j)$$

$$soil^{IWD} = soil^{IWD} + \Delta soil(i,j) \tag{13}$$

6. Find the iteration-best solution T^{IB} from all the solutions T^{IWD} found by the IWDs using

$$T^{IB} = arg \max_{\forall T^{IWD}} q(T^{IWD}) \tag{14}$$

where function $q(.)$ gives the quality of the solution.

7. Update the soils on the paths that form the current iteration-best solution T^{IB} by

$$soil(i,j) = (1+\rho_{IWD}).soil(i,j)$$
$$-\rho_{IWD}.\frac{1}{(N_{IB}-1)}.soil_{IB}^{IWD} \quad \forall(i,j) \in T^{IB} \tag{15}$$

where N_{IB} is the number of nodes in the solution T^{IB}.

8. Update the total best solution T^{TB} by the current iteration-best solution T^{IB} using

$$T^{TB} = \begin{cases} T^{TB} & q(T^{TB}) \geq q(T^{IB}) \\ T^{IB} & otherwise \end{cases} \tag{16}$$

9. Increment the iteration number by $Iter_{count} = Iter_{count} + 1$. Then, go to Step 2 if $Iter_{count} < Iter_{max}$.

10. The algorithm stops here with the total-best solution T^{TB}.

The application of IWD algorithm to ELD problem is shown in Fig. 1.

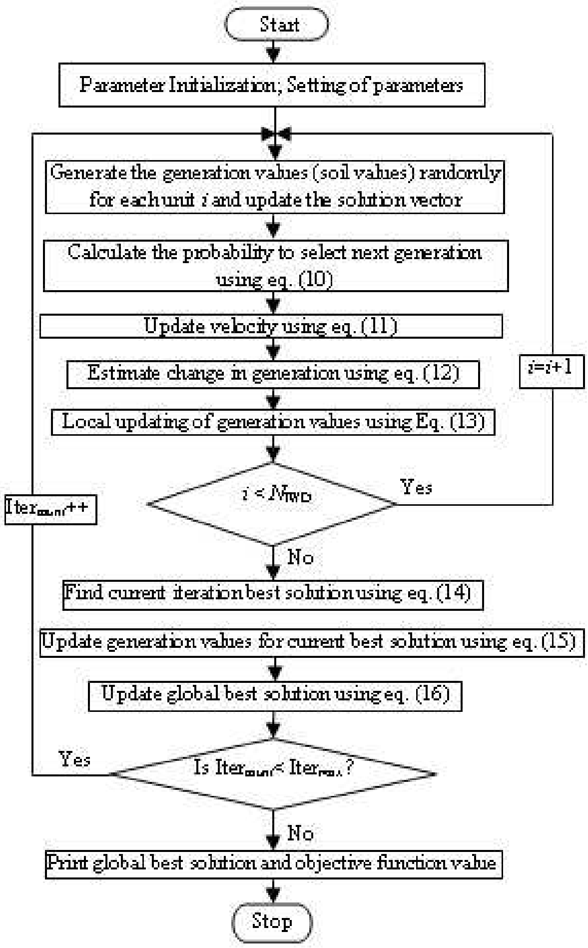

Fig. 1. Flowchart of the proposed IWD algorithm

C. Constraints handling

A key factor in the application of IWD approaches to the optimization of an EDP is how the this algorithm handles the constraints relating to the problem. The search space in constrained optimization problems consists of two kinds of points: feasible and unfeasible. Feasible points satisfy all the constraints, while unfeasible points violate at least one of them. Therefore, the solution or set of solutions obtained as the final result of an optimization method must necessarily be feasible, i.e., they must satisfy all constraints. The methods based on the use of penalty functions are usually employed to treat constrained optimization problems. A constrained problem can be transformed into an unconstrained one by penalizing the constraints and building a single objective function, which in turn is minimized using an unconstrained optimization algorithm. When optimization algorithms are used for constrained optimization problems, it is common to handle constraints [19] using concepts of penalty functions (which penalize unfeasible solutions), i.e., one attempt to solve an unconstrained problem in the search space *S* using a modified fitness function *f* (minimizing the fitness function in this paper) such as

$$Min.C_T = \begin{cases} C_i(P_i) & if P_i \in F \\ C_i(P_i) + \lambda(P_i) & otherwise \end{cases} \tag{17}$$

where $\lambda(P_i)$ is zero and no constraint is violated; otherwise it is positive. The penalty function is usually based on a distance measured to the nearest solution in the feasible region *F* or to the effort to repair the solution. In this work, the methodology used to constraint handling is divided into two steps. The first step involves finding solutions for the decision variables that lie within user-defined *upper* and *bounds*, that is, $x \in [lower, upper]$. Whenever a *lower* bound or an *upper* bound restriction fails to be satisfied, a repair rule is applied according to (18) and (19), respectively:

$$P_i^j(t+1) = P_i^j(t) + \beta.rand_i[0,1]\{upper(P_i^j(t)) \\ -lower(P_i^j(t))\} \tag{18}$$

$$P_i^j(t+1) = P_i^j(t) - \beta.rand_i[0,1]\{upper(P_i^j(t)) \\ -lower(P_i^j(t))\} \tag{19}$$

IV. Simulation Results

In order to assess effectiveness the proposed IWD algorithm is programmed in MATLAB environment and executed on a 3.2 GHz Pentium IV processor with 2 GB RAM. Two networks having 6 and 20 generators are simulated and test results are compared with popular methods reported in recent literature.

A. Six-unit system

This is a small test system consisting of six thermal units [20] and it has 26 buses and 46 transmission lines. The prohibited operating zones limits and generation limit constraints are considered in simulation. The required data of test system is given in Table I and Table II. The parameters of algorithm used for simulation are: Number of water drops

$N_{IWD} = 6$; Velocity updating parameters are a_v =1, b_v =0.01 and c_v = 1; Soil updating parameters as a_s =1, b_s =0.01 and c_s = 1. Local soil updating parameter, $\rho_n = 0.9$; Global soil updating parameter, $\rho_{IWD} = 0.9$; $InitSoil$ = 10000; $InitVel$ = 200; $Iter_{max}$=100.

TABLE I
CAPACITY AND COST COEFFICIENT DATA OF SIX UNIT THERMAL SYSTEM

Unit	P_i^{min} (MW)	P_i^{min} (MW)	a_i $\$/MW^2$	b_i $\$/MW$	c_i $\$$
1	500	100	0.0070	7	240
2	200	50	0.0095	10	200
3	300	80	0.0090	8.5	220
4	150	50	0.0090	11	200
5	200	50	0.0080	10.5	220
6	120	50	0.0075	12	120

TABLE II
PROHIBITED ZONE LIMITS OF SIX UNIT SYSTEM

Unit	Prohibited Zones	
	Zone 1	Zone 2
1	[210-240]	[350-380]
2	[90-110]	[140-160]
3	[150-170]	[210-240]
4	[80- 90]	[110-120]
5	[90-110]	[140-150]
6	[75- 85]	[100-105]

This system supplies a total load of P_D = 1263 MW. The B-matrix for transmission network is given as

$$B_{ij} = 10^{-3}.\begin{bmatrix} 1.7 & 1.2 & 0.7 & -0.1 & -0.5 & -0.2 \\ 1.2 & 1.4 & 0.9 & 0.1 & -0.6 & -0.1 \\ 0.7 & 0.9 & 3.1 & 0.0 & -1.0 & -0.6 \\ -0.1 & 0.1 & 0.0 & 2.4 & -0.6 & -0.8 \\ -0.5 & -0.6 & -1.0 & -0.6 & 12.9 & -0.2 \\ -0.2 & -0.1 & -0.6 & -0.8 & -0.2 & 15.0 \end{bmatrix}$$

$$B_{i0} = 10^{-3}.\begin{bmatrix} -0.39 & -0.13 & 0.71 & 0.06 & 0.22 & 0.66 \end{bmatrix}$$

$$B_{00} = \begin{bmatrix} 0.0056 \end{bmatrix}$$

Results obtained using the proposed method and results of PSO [20], GA [20], and BBO [16] are presented in Table III. To verify the performance of the proposed algorithm, this test case was repeatedly solved 100 times. The minimum, average, and maximum values of objective function are presented in Table III. The minimum value of cost function obtained for this test case using IWD is 15439 \$/h which is comparatively lower than other methods reported. The mean cost and standard deviation (STD) of objective function for 100 runs also provided in Table III. The mean value and STD of the proposed method are less than GA, PSO, and BBO algorithms. Further, a smaller value of standard deviation implies that most of the best solutions are close to the average value. The best solutions for these 100 runs are compared with best objective function values obtained by using PSO [20], GA [20], and BBO [16]. The results reported using PSO [20], GA [20], and BBO [16] got premature convergence so that their standard deviation is larger than that of IWD algorithm.

In order to assess speed of the algorithm, the proposed, PSO [20], GA [20], and BBO [16] algorithms are programmed and implemented on same platform. The CPU time of all algorithms are also presented in Table III. The CPU time required to obtained the minimum solution using 100 iterations by the proposed method is 2.54 sec, which is almost same as that of BBO but less by sixteen and six times that of GA and PSO. Therefore, the proposed method is able to find the near optimal solution with less computational time compare to GA, PSO, and BBO algorithms. A convergence characteristic of the proposed IWD algorithm for 6 unit thermal system is shown in Fig. 2. From the characteristic it is seen that solution is converged to near optimal solution after 48^{th} iteration.

TABLE III
COMPARISON OF BEST OUTPUTS OF 6-UNIT SYSTEM USING DIFFERENT METHODS

item	GA [20]	PSO [20]	BBO [16]	Proposed IWD Method
P_1 (MW)	474.81	447.50	447.4	450.13
P_2 (MW)	178.64	173.32	173.24	173.62
P_3 (MW)	262.21	263.47	263.32	260.61
P_4 (MW)	134.28	139.06	138	139.49
P_5 (MW)	151.9	165.48	165.41	159.70
P_6 (MW)	74.18	87.13	87.08	90.51
Total Power (MW)	1276.02	1275.96	1274.44	1274.05
P_{Loss} (MW)	13.02	12.96	12.44	12.05
Min. cost($\$/h$)	15459	15450	15443	15439
Avg. cost($\$/h$)	15469	15454	15449	15445
Max. cost($\$/h$)	15524	15492	15485	15461
STD	5.47	2.88	2.72	1.87
CPU Time(s)	41.58	14.86	3.25	2.54

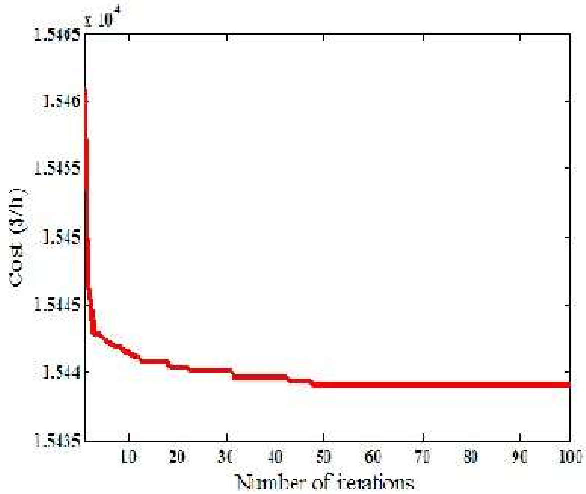

Fig. 2. Convergence characteristics of 6 unit thermal system

B. Twenty-unit system

This test system consists of twenty thermal units and supplies a total load of P_D = 2500 MW. The data of the system and B-matrix is available in [21]. All parameters of algorithm is same as test case 1 except $Iter_{max}$=200. The simulation results of the proposed method and PSO [20], BBO [16], Hop field model [21] approaches are summarized in Table IV.

Results obtained using the proposed method and results of PSO [20], BBO [16], and Hop field model [21] are presented in Table IV. To verify performance of the proposed algorithm, this test case was repeatedly solved 200 times. Optimum generation values each generator using the proposed method and other methods are given in Table IV. The minimum, average and maximum total costs obtained using the proposed method and other methods mentioned above are also reported in Table IV. The minimum value of cost function obtained for this test case using IWD is 59799 \$/h which is comparatively lower than other methods reported. From results, it is seen that total cost of generation of the proposed method is less than all other methods. The mean cost and standard deviation (STD) of objective function for 200 runs also provided in Table IV. A smaller standard deviation implies that most the best solutions are close to the average. Therefore, from results presented, it is seen that the proposed method is fastest among all methods mentioned. Similar to test case 1, all algorithms are programmed and implemented on same platform and CPU times of all algorithms are also presented in Table IV. The CPU time of the proposed method to get the best solution is less by 2 and 5 times compared to BBO and PSO.

TABLE IV
COMPARISON OF BEST OUTPUTS OF 20-UNIT SYSTEM USING DIFFERENT METHODS

item	PSO [20]	BBO [16]	Hopfield [21]	Proposed IWD Method
P_1 (MW)	563.32	513.09	512.78	563.32
P_2 (MW)	106.56	173.35	169.1	106.56
P_3 (MW)	98.71	126.92	126.89	98.71
P_4 (MW)	117.32	103.33	102.87	117.32
P_5 (MW)	67.08	113.77	113.68	67.08
P_6 (MW)	51.47	73.07	73.57	51.47
P_7 (MW)	47.73	114.98	115.29	47.73
P_8 (MW)	82.43	116.42	116.4	82.43
P_9 (MW)	52.09	100.69	100.41	52.09
P_10 (MW)	106.51	100	106.03	106.51
P_11 (MW)	197.94	148.98	150.24	197.94
P_12 (MW)	488.33	294.02	292.76	488.33
P_13 (MW)	99.95	119.58	119.12	99.95
P_14 (MW)	79.89	30.55	30.83	79.89
P_15 (MW)	101.53	116.45	115.81	101.53
P_16 (MW)	25.84	36.23	36.25	25.84
P_17 (MW)	70.02	66.86	66.86	70.02
P_18 (MW)	53.95	88.55	87.97	53.95
P_19 (MW)	65.43	100.98	100.8	65.43
P_20 (MW)	36.26	54.27	54.31	36.26
Total Power (MW)	2512.34	2592.11	2591.97	2512.34
P_{Loss} (MW)	92.33	92.11	91.97	92.33
Min. cost($\$/h$)	59804	62456.79	62456.63	59799
Avg. cost($\$/h$)	61171	62456.79	62493.05	61151
Max. cost($\$/h$)	63184	62456.79	62517.84	63176
STD	532.44	–	–	529.16
CPU Time(s)	15.3	6.93	–	3.9

A convergence characteristic of IWD algorithm for 20 unit thermal system is shown in Fig. 3. From the characteristic it is seen that solution is converged to near optimal solution after 90^{th} iteration.

V. CONCLUSION

In this paper a novel approach based on Intelligent Water Drops (IWD) algorithm to solve economic load dispatch problem, considering various generator constraints, has been sucessfully applied. The feasibility of the proposed algorithm

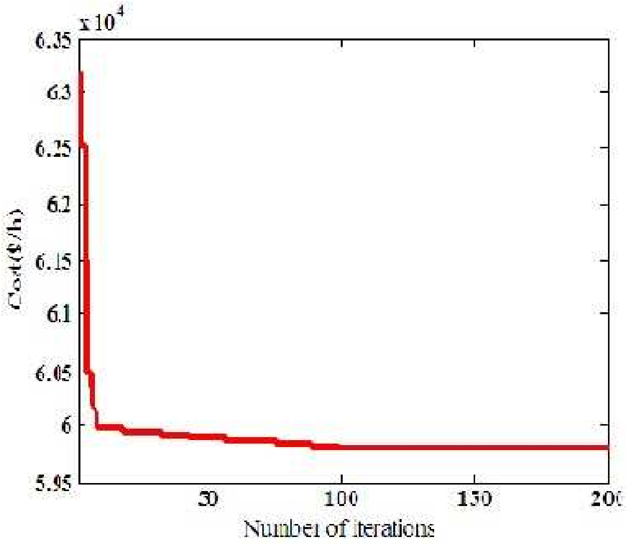

Fig. 3. Convergence characteristics of 20 unit thermal system

for solving ELD problem is demonstrated using 6-unit and 20-unit thermal systems. Moreover, in order to handle constraints effectively, a constraint treatment mechanism inspired in [19] is devised in the calculation of cost function used. Numerical results reveal that the proposed algorithm converged to good solutions in comparison with results obtained using GA, PSO, BBO, Hopfiled approaches with less computational effort. Further, the standard deviation of the proposed method is less than other approaches. This shows all the good solutions are closed to average values and hence the algorithm is robust and has fast convergence compared to methods mentioned.

Although the proposed algorithm had been successfully applied to ELD with valve-point loading effect and included a few constraints, the practical ELD problems should consider multiple fuels as well as spinning reserve and ramp rate constraints. This remains a challenge for future work.

REFERENCES

[1] A. J. Wood and B. F. Wollenberg, "Power generation operation and control", *John Wiley and Sons, New York*, 1984
[2] J. B. Park, K. S. Lee, J. R. Shin and K. Y. Lee, "A particle swarm optimization for economic dispatch with non smooth cost functions", *IEEE Trans. on Power Systems*, Vol. 8, No. 3, pp. 1325-1332, Aug. 1993.
[3] H. T. Yang, P. C. Yang and C. L. Huang, "Evolutionary Programming Based Economic Dispatch For Units With Non-smooth Fuel Cost Functions", *IEEE Transactions on Power Systems*, Vol. 11, No. 1, pp. 112-118, 1996.
[4] T. Jayabarathi, G. Sadasivam and V. Ramachandran, "Evolutionary programming based economic dispatch of generators with prohibited operating zones", *Electric Power Systems Research*, Vol. 52, No. 3, pp. 261-266, 1999.
[5] Z. X. Liang and J. D. Glover, "A zoom feature for a dynamic programming solution to economic dispatch including transmission losses, *IEEE Trans. on Power Systems*, Vol. 7, No. 2, pp. 544-550, May 1992.
[6] El-Sharkawy, M., and Nieebur, D., "Artificial neural networks with application to power systems", *IEEE Power Engineering Society, A Tutorial Course*, 1996.
[7] Yalcinoz, T., and Short, M. J., "Neural networks approach for solving economic dispatch problem with transmission capacity constraints", *IEEE Trans. on Power Systems*, Vol. 13, pp. 307-313, 1998.

[8] K.Y. Lee, A. Sode-Yome, J.H. Park, “Adaptive hopfield neural networks for economic load dispatch”, *IEEE Trans. on Power Systems*, Vol.13, No. 2, pp. 519526, 1998.
[9] J.H. Park,Y.S. Kim, I.K. Eom, K.Y. Lee, “Economic load dispatch for piecewise quadratic cost function using Hop field neural network”, *IEEE Trans. on Power Systems*, Vol. 8, No. 3, pp. 10301038, 1993.
[10] Pancholi, R. K., and Swarup, K. S., “Particle swarm optimization for security constrained economic dispatch”, *International Conference on Intelligent Sensing and Information Processing, Chennai, India*, pp. 712, 2004.
[11] Youssef, H. K., and El-Naggar, K. M., “Genetic based algorithm for security constrained power system economic dispatch”, *Electric Power Systems Research*, Vol. 53, pp. 4751, 2000.
[12] S.O. Orero, M.R. Irving, “Economic dispatch of generators with prohibited operating zones: a genetic algorithm approach”, *IEE Proc. Gen. Transm. Distrib.*, Vol. 143, No. 6, pp. 529534, 1996.
[13] D. C. Walters and G. B. Sheble, “Genetic algorithm solution of economic dispatch with the valve-point loading”, *IEEE Trans. on Power Systems*, Vol. 8, No. 3, pp. 1325-1332, Aug. 1993.
[14] Nasimul Nomana, Hitoshi Iba, “Differential evolution for economic load dispatch problems”, *Electric Power Systems Research*, Vol. 78, pp. 13221331, 2008.
[15] W. M. Lin, F. S. Cheng and M. T. Tsay, “An improved Tabu search for economic dispatch with multiple minima”, *IEEE Trans. on Power Systems*, Vol. 17, No. 1, pp.108-112, Feb. 2002.
[16] Aniruddha Bhattacharya, P.K. Chattopadhyay, “Solving complex economic load dispatch problems using biogeography-based optimization”, *Expert Systems with Applications*, Vol. 37, pp. 36053615, 2010.
[17] Hamed Shah-Hosseini, “The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm”, *International Journal of Bio-Inspired Computation*, Vol. 1, Nos. 1 and 2, pp. 71-79, 2009.
[18] Shah-Hosseini. H, “Optimization with the Nature- Inspired Intelligent Water Drops Algorithm”, *Int. Journal of Intelligent Computing and Cybernetics*, Vol. 1, No. 2, pp. 193-212, 2008.
[19] [25] Z. Michalewicz, M. Schoenauer, “Evolutionary algorithms for constrained parameter optimization problems”, Evol. Comput. vol. 4, no. 1, pp. 1-32, 1996.
[20] Gaing ZL, “Particle swarm optimization to solving the economic dispatch considering the generator constraints”, *IEEE Trans. on Power Systems*, Vol. 18, No. 3, pp. 118795, 2003.
[21] Su CT, Lin CT, “New approach with a Hop field modeling framework to economic dispatch”, *IEEE Trans. on Power Systems*, Vol. 15, No. 2, pp. 541545, 2000.

Biography

R. Srinivasa Rao received the B.Tech in Electrical Engineering degree from SVU Tirupathi (INDIA)in 1996, the M.E degree from IISc Bangalore in 1998, and the Ph.D. degree from JNTU Hyderabad in 2009. Currently, he is Associate Professor in department of electrical engineering, JNTU Kakinada, INDIA. His areas of interest include electric power distribution systems and power systems operation and control.