Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 6, No. 4, November 2015), Pages: 13-26
www.jacr.iausari.ac.ir

Application of Swarm-Based Optimization Algorithms for Solving Dynamic Economic Load Dispatch Problem

Alireza Khosravi[1], **Mohammad Yazdani-Asrami**[✉2]

1) Faculty of Electrical and Computer Engineering, Babol University of Technology,Babol,Iran
2) Young Researchers and Elite Club, Sari Branch, Islamic Azad University, Sari, Iran

akhosravi@nit.ac.ir; yazdani@stu.nit.ac.ir

Received: 2014/08/04; Accepted: 2014/12/02

Abstract

Dynamic economic load dispatch is one of the most important roles of power generation's operation and control. It determines the optimal controls of production of generator units with predicted load demand over a certain period of time. Economic dispatch at minimum production cost is one of the most important subjects in the power network's operation, which is a complicated nonlinear constrained optimization problem. Since dynamic economic load dispatch was introduced, several intelligent methods have been used to solve this problem. In this paper, an Improved Particle Swarm Optimizer (IPSO) and Water Cycle optimizer (WCO), as swarm-based optimization algorithms, have been proposed to solve dynamic economic load dispatch problem and their results compare with each other. These algorithms are applied to a dynamic economic dispatch problem for 6-unit power systems with a 24-h load demand at each one hour time intervals. The goal of the research is categorized in two parts; first of all, introduction of application of new heuristic method for solving economic load dispatch problem and second, comparison between two swarm-based algorithms. Obtained results show that WCO is very fast and also reach to better results and minimum.

Keywords: *Dynamic Economic Load Dispatch, Improved Particle Swarm Algorithm, Power Loss, Water Cycle Algorithm.*

1. Introduction

Dynamic economic load dispatch is an extension of static economic load dispatch to determine the generation schedule of the committed units so as to meet the predicted load demand over a time horizon at minimum operating cost under valve point, ramp rate, multi fuel and other constraints. The dynamic economic load dispatch is a method to schedule the online generator outputs with the predicted load demands over a certain period of time, so as to operate an electric power system most economically. It is a dynamic optimization problem taking into account the constraints imposed on system operation by generator ramping rate limits. The dynamic economic load dispatch is not only the most accurate formulation of the economic dispatch problem but also the most difficult to solve because of its large dimensionality. Normally, it is solved by dividing the entire dispatch period into a number of small time intervals, and then a static economic dispatch has been employed to solve the problem in each interval. Since dynamic economic load dispatch was introduced, several methods have been used to solve this problem. However, all of those methods may not be able to provide an optimum solution and usually getting stuck at local optima [1-3]. Depending on the complexity of the problem, different objective functions are defined as ‘cost functions’. Some of these functions are modeled as linear functions and more complex ones are modeled as non-linear functions. Solutions with lower operation cost are more willing to reach the optimized operation point. So, engineers always attempt to decrease the operation cost with different mathematical techniques.

Many mathematical methods have been developed to solve the dynamic economic load dispatch problem in the past decades. The major methods include LP [4], NLP [5], LRA [6] and QP [7]. These methods were facing problems to give optimal solution due to the non-linear and non-convex characteristics of generating units. It would generate large errors to use LP to linearize the dynamic economic load dispatch model; also, for QP and NLP; the objective function should be continuous and differentiable. The LR algorithm leads to the solution oscillation. Dynamic programming is a method that can solve the dynamic economic load dispatch problem without imposing any restrictions on the nature of the cost curves. However, this method suffers from the dimensionality, leading to high computational cost.

Modern heuristics stochastic optimization techniques such as PSO [8, 9], TS [10], GA [11, 12], SA [13], HNN [14, 15], and EP [16, 17] appear to be efficient in solving dynamic economic load dispatch problem without any restriction on the shape of cost curves due to their ability to seek the optimal solution. In addition, as a very new research, a new algorithm called Brent Method was proposed to solve a problem dynamically in [18]. Also in [19, 20], conventional economic load dispatch with considering valve-point effect was solved with GSA.

The most important issue with evolutionary techniques is to maintain a proper balance between exploration i.e. global search and exploitation i.e. local search. The performance of evolutionary methods heavily depends on the settings of the tuning parameters; therefore finding optimal parameter setting is a very big challenge. Evolutionary methods also have a tendency to converge very fast to a solution that is quite close to the global minimum. This tendency causes premature convergence.

In this paper, new methods based on swarm intelligence are used for dynamic economic load dispatch. In this method, a new version of PSO and also WCO are presented. To combine these algorithms with economic load dispatch problem, number of generation units is considered as problem dimension. In different iteration, power outputs of generation units are calculated with respect to system constraints. One unit is selected randomly as a "slag generator" to compensate the difference between load demand and total power generation. The effectiveness of the presented method is demonstrated for a 6-unit test system. Also, simulation results have been compared with reported results in literatures and demonstrate that solving dynamic economic load dispatch problem with WCO and IPSO lead to very accurate and better results.

The paper is organized as follow: sections 2 emphasize on the dynamic economic load dispatch problem formulation with various constraints.Section 3, gives a brief description about the IPSO algorithm. Also, in section 4, a brief description about the WCO algorithm was presented. Section 5, presents the implementation of swarm-basedalgorithms for dynamic economic load dispatch problem and also, simulation results for test case are compared with the other reported methods.

2. Dynamic Economic Load Dispatch Problem Formulation

Dynamic economic load dispatch is an important topic in power system operation and many papers published in literature about the problem. In each sub-time interval, total power generation is adjusted to supply the required demand in a minimum operation cost, which optimizes the production of the costly fossil generation units.

A simple dynamic economic load dispatch problem consists of some continuous functions which can be solved by mathematical techniques. Dynamic economic load dispatch problem includes physical and operational constraints that are described as equality and inequality constraints. Several of these constraints have been introduced as follow [1-5]:

Equality constraints (without loss):

$$\sum_{i=1}^{n} P_{i,t} = P_{D,t} \qquad (1)$$

Inequality constraints:

$$P_i^{\min} \le P_i \le P_i^{\max} \qquad (2)$$

Where $P_{i,t}$ is the output power of i_{th} generator in the t_{th} time interval in MW, $P_{D,t}$ is the total power demand in the t_{th} time interval in MW, P_i^{min} is the lower bound and P_i^{max} is the upper bound of the generation of the i_{th} unit in MW and also, n is the total number of power units.

In more complex problems, transmission loss (P_L) is added to (1). The P_L is calculated as follow:

$$P_L = \sum_{i=1}^{n}\sum_{j=1}^{n} P_i B_{ij} P_j + \sum_{i=1}^{n} B_{0i} P_i + B_{00} \qquad (3)$$

Calculations of B-coefficients are described in [2-7]. Therefore, with substituting (3) in (1), (4) is deduced:

$$\sum_{i=1}^{n} P_i = P_D + P_L \qquad (4)$$

Fuel cost of each generator is defined as:

$$F_i(P_i) = a_i P_i^{2} + b_i P_i + c_i \qquad (5)$$

Where F_i is the fuel cost and a_i, b_i and c_i are cost coefficients of i_{th} generator.

All of the aforementioned cost functions are continuous functions and can be solved by conventional mathematical techniques. More accurate models include ramp-rate constraints which are modeled as [1-5]:

a) As generation increases:

$$P_i - P_i^0 \le UR_i \qquad (6)$$

b) As generation decreases:

$$P_i^0 - P_i \le DR_i \qquad (7)$$

So, equation (2) can be corrected as:

$$Max\,(P_i^{\min}, P_{i,t}^0 - DR_i) \le P_{i,t} \le \min(P_i^{\max}, P_{i,t}^0 + UR_i) \quad (8)$$
$$i = 1, 2, \ldots to\, n \quad t = 1, 2, \ldots T$$

Where UR_i and DR_i are up- ramp rate and down-ramp rate limits of the i_{th} generator, respectively.

Finally, minimizing cost function is the purpose of solving ELD problems:

$$\min f = \sum_{t=1}^{T} \sum_{i=1}^{n} F_{i,t}(P_{i,t}) \quad (9)$$

3. Concept of IPSO

Kennedy and Eberhart suggested a PSO based on the analogy of swarm of bird and school of fish. In PSO, each individual makes its decision based on its own experience together with other individual's experiences [21]. The individual particles are drawn stochastically towards the position of present velocity of each individual, their own previous best performance, and the best previous performance of their neighbors. It was developed through simulation of a simplified social system, and has been found to be robust in solving continuous non-linear optimization problems [22]. The main advantages of the PSO algorithm are summarized as: simple concept, easy implementation, and computational efficiency when compared with mathematical algorithm and other heuristic optimization techniques [23-25].

The PSO follows the special scenario: suppose that a group of birds are randomly searching food in a big area. Suddenly one of them finds a piece of food. Other birds don't know where the food is, but they know which finds the food and how far from it. So, the best strategy is to follow the bird which is nearest to the food [21]. Using this scenario, PSO can be used to solve optimization problems. In PSO, each single solution is a ''bird'' in the search space. Here, it is called as ''particle''. For all of the particles fitness value has been calculated, which are evaluated by the fitness function to be optimized, and have velocities, which direct the flying of the particles. The particles are ''flown'' through the problem space by following the current optimum particles. The PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In each iteration, each particle is updated by following two ''best'' values. The first one is the best solution (fitness) it has achieved so far; and he fitness value is also stored. This value is called "Pbest". Another ''best'' value that is tracked by the PSO is the best value, obtained so far by any particle in the population which is a global best that is called "Gbest" [21-23]. After finding the two best values, the particle updates its velocity and positions based on following equations:

$$V_{t+1} = W \times V_t + C_1 \times r \times (P_{pb} - X_{cs}) + C_2 \times r \times (P_{gb} - X_{cs}) \quad (10)$$

$$X_{t+1} = X_t + V_{t+1} \quad (11)$$

The main PSO categorized into two major topologies: global and local search PSO. Several tests show that global version has a worse search space coverage than the local PSO, also in global version particles' movement are more concentrated around one solution and therefore can more quickly find the best solution. A little problem may occur for global version is very susceptible to local minima. In general, global version is the better choice when solution space is not very scattered, because of its speed and accuracy [26, 27].

Fig.1 shows the concept of the searching mechanism of PSO using the modified velocity and position of each individual by (10) and (11).

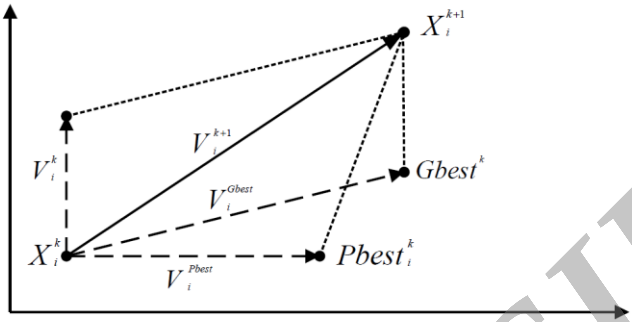

Figure 1. The search mechanism of the PSO

Following code shows global PSO algorithm's pseudo code:

```
    For each particle
        Initialize particle
    End
Do
    For each particle
        Evaluate objective function and calculate fitness value
        If the fitness value is better than the best fitness value Pbest in history
        Set current value as the new Pbest
    End
Choose the particle with the best fitness value of all the particles as the Gbest
    For each particle
        Update particle's velocity
        Update Particle's position
    End
While maximum iteration or minimum error criteria is not attained
```

Inertia weight is the other factor that its changes can make new version of PSO. In this case, amount of momentum that a particle carries between iteration can be controlled by a parameter *w* that multiplied by the particle's current velocity, as it can be seen in equation (10). Indeed, this control parameter influences the particle's area of exploration. So, the amount of this parameter can play an important role in searching process. Using a high constant value of inertia weight (e.g. more than unit) although can cover more area, but often PSO trap in local minima and can't find best solution. In the other hand, a lower value of this parameter will lead to particles concentrating on small search space and PSO losses the other possible solution spaces. One idea is using an equation for inertia weight that depends on the number of pass iterations so that decreases during search process. Using high value of inertia weight in initialization step can help PSO to search all area of the solution space, by spending time and increase of the number of iteration, inertia weight decreases; this can help PSO to present good local search at the final iterations [21-27].

The formula for inertia weight can be exponential format, as shown in equation (12). Using this equation, PSO starts by a given value and decreases exponentially to near zero.

$$W = w_{\max} e^{-\alpha \frac{t}{T}} \tag{12}$$

Where, w_{max} is the maximum values of inertia weight. The coefficient α is a positive constant that amount of it can control ramp of variations.

In this paper, an improved version of PSO is utilized. This version uses constant value for inertia weight but this constant is calculated based on equation (13). It is not a simple equation, while it is obtained by analytical and algebraic analysis that uses constriction property. Constriction coefficients can prevent explosion; further, these coefficients can induce particles to converge on local optima [26, 27]. In addition in this IPSO, equation (10) has a significant change that can be seen equation (13):

$$\begin{cases} \chi = \dfrac{2\kappa}{\varphi - 2 + \sqrt{\varphi^2 - 4\varphi}} \; for\, \varphi > 4 \\ else\; k \end{cases} \tag{13}$$

$$V_{t+1} = \chi \times (V_t + \varphi_1 \times (P_{pb} - X_{cs}) + \varphi_2 \times (P_{gb} - X_{cs})) \tag{14}$$

Where, coefficient χ is constriction coefficient which is used as inertia weight. Coefficients φ_1 and φ_2 are random numbers uniformly distributed in the range (0, φ/2). Set k=1, meaning that the space thoroughly searched before the swarm collapses into a point [26, 27].

4. Concept of WCO

This novel heuristic algorithm introduced at 2012, is taken from the behavior of water cycling in nature. Water moves downhill in the streams and rivers, starting from up in the mountains and ending up in the sea. Streams and rivers collect water from the rain and other streams on their way. The rivers and lakes are evaporated when plants give off water as transpire process. Then, by carrying the water in the atmosphere, clouds will be generated. These clouds condense in the colder atmosphere and release the water back in the rain form, creating new streams and rivers. Fig. 2 shows the schematic procedure of the WCO [28].

Like other swarm-based algorithms, this method begins with an initial population called raindrops caused by rain or precipitation. The best raindrop is chosen as sea, a number of better raindrops as rivers and the rest, are considered as streams flowing to rivers or directly to the sea. In Water Cycle Algorithm (WCA), each array is considering N_{pop} individuals; the raindrops matrix is as follows:

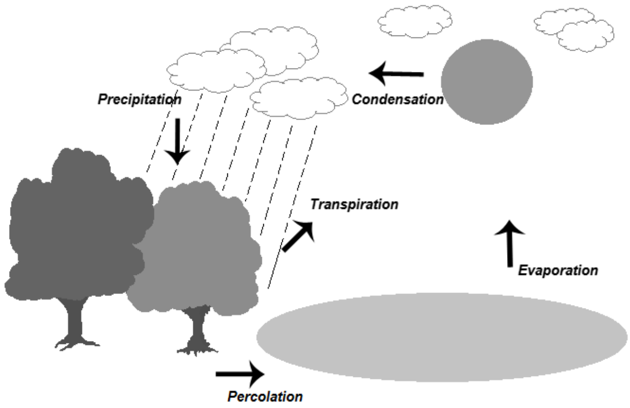

Figure 2. The Schematic of the WCO

$$\text{Population of raindrops} = \begin{bmatrix} \chi_1^1 & \chi_2^1 & \chi_3^1 & \cdots & \chi_{N\,\mathrm{var}}^1 \\ \chi_1^2 & \chi_2^2 & \chi_3^2 & \cdots & \chi_{N\,\mathrm{var}}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \chi_1^{Npop} & \chi_2^{Npop} & \chi_3^{Npop} & \cdots & \chi_{N\,\mathrm{var}}^{Npop} \end{bmatrix} \quad (15)$$

Where N_{pop} is the number of raindrops and N_{var} defines number of variables. In a randomly generated matrix of raindrops with the size of $N_{pop} \times N_{\mathrm{var}}$, Each of the decision variable values $(\chi_1, \chi_2, \chi_3, ..., \chi_N)$ can be represented as real values or as a predefined set for continuous and discrete problems, respectively. The fitness or cost of each row is obtained using the Cost function (C) given as:

$$C_i = Cost_i = f(x_1^i, x_2^i, ..., x_{N\,\mathrm{var}}^i) \quad (16)$$

$$i = 1, 2, 3, ..., N_{pop}$$

After generating N_{pop} raindrops, a number of N_{sr} among the best of them are chosen as rivers and sea. The raindrop which has the best function value is considered as sea. The rest, are considered as streams that may flow to the rivers or directly to the sea [28, 29]:

$$N_{sr} = \text{Number of Rivers} + \underbrace{1}_{Sea} \quad (17)$$

$$N_{Streams} = N_{pop} - N_{sr} \quad (18)$$

Depending on the intensity of the flow, Streams are assigned to the rivers and sea, which is calculated with the equation below:

$$NS_n = round\left\{ \left| \frac{Cost_n}{\sum_{i=1}^{NS_r} Cost_i} \right| \times N_{Streams} \right\} \quad (19)$$

Where, NS_n is the number of streams which flow to a specific river or sea.

The movement of a stream's flow toward a specific river is applied along the connecting line between them by using a randomly chosen distance as $X \in (0, C \times d)$. Where, C is a user-defined value between 1 and 2; and d is the current distance between stream and river. The value X is a number between 0 and $C \times d$ with any distribution. If the value of C be greater than 1, the streams gain ability to flow in different directions toward the rivers. So, the best value for C may be chosen as 2. This concept can be used in flowing rivers to the sea. Therefore, new position for streams and rivers can be calculated using [28-30]:

$$X_{Stream}^{i+1} = X_{Stream}^{i} + rand \times C \times \left(X_{River}^{i} - X_{Stream}^{i} \right) \quad (20)$$

$$X_{River}^{i+1} = X_{River}^{i} + rand \times C \times \left(X_{Sea}^{i} - X_{River}^{i} \right) \quad (21)$$

Where rand is a uniformly distributed random number between [0,1]. If any streams solution value is better than its connecting river, their position is changed. Also, the position of sea and a river is changed if the river has a better solution than the sea.

The evaporation process has an important role in the WCA, preventing the algorithm from trapped in local optima and rapid convergence. The following clause represents the determination of whether or not the evaporation and raining process happens.

$$if \left| \chi_{Sea}^{i} - \chi_{River}^{i} \right| < d_{\max} \quad i = 1,2,3,\ldots N_{sr} - 1 \quad (22)$$

Where, $d_{\max}$ is a small number close to zero and controls the search depth, near the sea. When a large value of $d_{\max}$ is selected, the search intensity is being reduced but its small value encourages it. When the distance between the river and sea is less than $d_{\max}$ the river has joined the sea. So, the evaporation process is applied and then the raining process will happen. The value of d decreases at the end of each iteration with equation below [28, 29]:

$$d_{\max}^{i+1} = d_{\max}^{i} - \frac{d_{\max}^{i}}{\text{max iteration}} \quad (23)$$

The new randomly generated raindrops form new streams in different locations. Again the raindrop with the best function value among other new raindrops is considered as a river flowing to the sea. The rest of them are considered as new streams which flow to the river or go directly to the sea. For the streams that directly flow to the sea a specific equation which increases the exploration near sea is used, result in improvements in the convergence rate and computational performance of the algorithm for constrained problems [28-30].

$$\chi_{stream}^{new} = \chi_{sea} + \sqrt{U} \times randn(1, N_{\text{var}}) \quad (24)$$

Where U defines the concept of variance. In fact the value of U shows the range of searching region near the sea and *randn* is a normally distributed random number. The most suitable value found for U is 0.1, while the higher values increases the possibility of quitting from feasible region and the lower values reduce the searching space and exploration near the sea.

5. Simulation Results and Discussion

In this paper, two swarm-based heuristic optimization algorithms have been utilized to minimize total fuel cost of generation units. Their implementation for dynamic economic load dispatch problem includes following steps:

Step (1):

Initialize Step: Initialize number of populations, Dimension of search space (number of generators), and total required demand in each time interval ($P_{D,t}$).

In this step, total required demand (P_D) and transmission loss (P_L) should be considered, as total generation output (P_D+P_L) should be satisfied.

$$\sum_{j=1}^{n} m_{i\,j} = P_D + P_L \tag{25}$$

Step (2):

Determination of generator limits and cost function coefficients: Generation units have some limits in producing power. So, these limits should be considered in dynamic economic load dispatch problems. Following equation indicates generation limits of i^{th} unit with respect to its ramp-rate constraints:

$$\max(m_i^{\min}, m_{i,t}^0 - DR_i) \le m_{i,t} \le \min(m_i^{\max}, m_{i,t}^0 + UR_i) \tag{26}$$

Step (3):

Random Generation of initial population: To start algorithms, population should be initialized within their feasible regions. In this paper, feasible region is defined between minimum and maximum power output of each generator.

Step (4):

Cost calculation for each agent: Here, generation cost of each unit is calculated using the proposed algorithm. Also, problem constraints are satisfied in the following manner:

$$\begin{aligned} &if \quad m_{i,t} \le \max(m_i^{\min}, m_{i,t}^0 - DR_i), \\ &then \quad m_{i,t} = \max(m_i^{\min}, m_{i,t}^0 - DR_i); \end{aligned} \tag{27}$$

And:

$$\begin{aligned} &if \quad \min(m_i^{\max}, m_{i,t}^0 + UR_i) \le m_{i,t}, \\ &then \quad \min(m_i^{\max}, m_{i,t}^0 + UR_i) = m_{i,t}; \end{aligned} \tag{28}$$

Step (5):

Updating algorithms parameters: Position and velocity should be updated for both algorithms.

Step (6):

Checking stop criteria: Stop criteria should be checked in this step.

Step (7):

Updating power output of each generator: After calculation of minimum cost, power output of each unit should be recomputed and defined as an initial value for the next time interval.

Step (8):

Implementing algorithms for all time intervals: If economic load dispatch is employed for each time interval, results (outputs) should be a scheduled program for generating power in a power system.

In this section, WCO and IPSO have been applied to solve dynamic economic load dispatchproblem in a 6-unit test case and the results obtained by them have been compared with other methods' results reported in literatures.

Cost coefficients and boundary limits of generation units are shown in Table 1. Ramp-rate limit constraints and initial power generated by each unit is shown in Table 2. Obviously, initial values for other time-interval are the values of the previous interval. Also, Table 3 shows the demand power for a 24-h time horizon at each 1-h time intervals.

Loss coefficients matrices that are related to equation (3) are shown below for WCO:

$$B_{ij} = 10^{-3} \begin{bmatrix} 1.7 & 1.2 & 0.7 & -0.1 & -0.5 & -2.0 \\ 1.2 & 1.4 & 0.9 & 0.1 & -0.6 & -0.1 \\ 0.7 & 0.9 & 3.1 & 0.0 & -1.0 & -0.6 \\ -0.1 & 0.1 & 0.0 & 0.24 & -0.6 & -0.8 \\ -0.5 & -0.6 & -0.1 & -0.6 & 12.9 & -0.2 \\ -2.0 & -1.0 & -0.6 & -0.8 & -0.2 & 15.0 \end{bmatrix} \tag{29}$$

$$B_{oi} = 10^{-3} \begin{bmatrix} -0.3908 & -0.1297 & 0.7047 & 0.0591 & 0.2161 & -0.6635 \end{bmatrix} \tag{30}$$

$$B_{oo} = 0.056 \tag{31}$$

Table 1. Generator Data and Cost Coefficients of a 6-Unit System

Unit	Minimum Power (Mw)	Maximum Power (Mw)	a ($/Mw)	b ($/Mw)	c ($/Mw)
1	100	500	0.007	7	240
2	50	200	0.00095	10	200
3	80	300	0.009	8.5	220
4	50	150	0.009	11	200
5	50	200	0.008	10.5	220
6	50	120	0.0075	12	190

Table 2. Ramp Rate Limits of the Studied 6-Unit System

Unit	Initial power generated(Mw)	Up-Ramp Rate (Mw/h)	Down-Ramp Rate (Mw/h)
1	340	80	120
2	134	50	90
3	240	65	100
4	90	50	90
5	110	50	90
6	52	50	90

Table 3. Demand Data with 24-h Time Horizon for the Studied 6-Unit System

Hour	Demand (Mw)	Hour	Demand (Mw)
1	955	13	1190
2	942	14	1251
3	935	15	1263
4	930	16	1250
5	935	17	1221
6	963	18	1202
7	989	19	1159
8	1023	20	1092
9	1126	21	1023
10	1150	22	984
11	1201	23	975
12	1235	24	960

Simulation results for a test case have been compared with other method's results in Table 4. The goal of the research is categorized in two parts; first of all, introduction of application of new heuristic method for solving economic load dispatch problem and second, comparison between two swarm-based algorithms. Comparative studies show that for a dynamic economic load dispatchproblem, WCO is faster and more accurate in reaching good solutions and can perform better than other methods.

Table 4. IPSO and WCO Results for dynamic economic load dispatch problem for a 6-Unit Test System

Applied Method	Minimum Cost ($/h)
Lambda Iterative Method	313405.648
Brent Method	313405.403
IPSO	313401.426
WCO	313399.721

6. Conclusion

Dynamic economic load dispatch is a problem to schedule the online generator outputs with the predicted load demands over a certain period of time, to operate an electric power system most economically. It is a dynamic optimization problem taking into account the constraints imposed on system operation by generator ramping rate limits. The dynamic economic load dispatch is not only the most accurate formulation of the economic dispatch problem but also the most difficult to solve because of its large dimensionality. This paper presents application of two swarm-based heuristic algorithms which are called WCO and IPSO and then, applied them for solving dynamic economic load dispatch problem. To show the efficacy of WCO in solving dynamic economic load dispatch problem, it is tested a 6-unit power systems. In comparison to other method's results, WCO has the ability to solve the optimization problem in a shorter time.

References

[1] X. S. Han, H. B. Gooi, and D. S. Kirschen, Dynamic economic dispatch: Feasible and optimal solutions, IEEE Transactions on Power Systems, 2001, 16(1), pp. 22-28.

[2] R. A. Jabr, A. H. Coonick, and B. J. Cory, A study of the homogeneous algorithm for dynamic economic with network constraints and transmission losses, IEEE Transactions on Power Systems, 2000, 15(2), pp. 605-611.

[3] P. Attaviriyanupap, H. Kita, E. Tanaka, and J. Hasegawa, A Hybrid EP and SQP for Dynamic Economic Dispatch With Nonsmooth Fuel Cost Function, IEEE Transactions on Power Systems, 2002, 17(2), pp. 411-416.

[4] A. Jabr, H. Coonick, and J. Cory, A homogeneous linear programming algorithm for the security constrained economic dispatch problem, IEEE Transactions on Power Systems, 2000, 15(3), pp. 930-937.

[5] C. Chen, Non-convex economic dispatch: a direct search approach, Energy Conversion and Management, 2007, 48(1), pp. 219-225.
[6] A. Keib, H. Ma, and J. Hart, Environmentally constrained economic dispatch using the Lagrangian relaxation method, IEEE Transactions on Power Systems, 1994, 9(4), pp. 1723-1730.
[7] L. Papageorgiou, and E. Fraga, A mixed integer quadratic programming formulation for the economic dispatch of generators with prohibited operating zones, Electric Power System Research, 2007, 77(10), pp. 1292-1298.
[8] X. Yuan, A. Su, Y. Yuan, H. Nie, and L. Wang, An improved PSO for dynamic load dispatch of generators with valve-point effects, Energy, 2009, 34(1), pp. 67-74.
[9] Z. Gaing, Particle swarm optimization to solving the economic dispatch considering the generator constraints, IEEE Transactions on Power Systems, 2003, 18(3), pp. 1187-1195.
[10] S. Pothiya, I. Ngamroo, and W. Kongprawechnon, Application of multiple tabu search algorithm to solve dynamic economic dispatch considering generator constraints, Energy Conversion and Management, 2008, 49(4), pp. 509-516.
[11] S. Baskar, P. Subbaraj, and M. Rao, Hybrid real coded genetic algorithm solution to economic dispatch problem, Computers and Electrical Engineering, 2003, 29(3), pp. 407-419.
[12] F. Li, and R. K. Aggarwal, Fast and accurate power dispatch using a relaxed genetic algorithm and a local gradient technique, Expert System with Applications, 2000, 19(3), pp. 159-165.
[13] C. K. Panigrahi, P. K. Chattopadhyay, R. N. Chakrabarti, and M. Basu, Simulated annealing technique for dynamic economic dispatch, Electric Power Components and Systems, 2006, 34(5), pp. 577-586.
[14] K. Swarup, and P. Simi, Neural computation using discrete and continuous Hopfield networks for power system economic dispatch and unit commitment, Neurocomputing, 2006, 70(1), pp. 119-129.
[15] S. Balakrishnan, P. Kannan, and C. Aravindan, On-line emission and economic load dispatch using adaptive Hopfield neural network, Applied Soft Computing, 2003, 2(4), pp. 297-305.
[16] N. Sinha, R. Chakrabarti, and P. Chattopadhyay, Evolutionary programming technique for economic load dispatch, IEEE Trans Evolutionary Computing, 2003, 7(1), pp. 83-94.
[17] A. Pathom, K. Hiroyuki, and T. Eiichi, A hybrid EP and SQP for dynamic economic dispatch with nonsmooth fuel cost function, IEEE Transactions on Power Systems, 2002, 17(2), pp. 411-416.
[18] K. Chandram, N. Subrahmanyam, and M. Sydulu, Brent method for Dynamic Economic Dispatch with Transmission Losses, Iranian Journal of Electrical and Computer Engineering, 2009, 8(1), pp. 16-22.
[19] S. Duman, U. Guvenc, and N. Yorukeren, Gravitational Search Algorithm for Economic Dispatch with Valve-Point Effects, International Review of Electrical Engineering, 2010, 5(6), pp. 2890-2895.
[20] H. Maskani, M. Yazdani-Asrami, M. Taghipour, A. Darzi, A. Moradi, and H. Falaghi, Gravitational Search Algorithm Optimization for Economic Dispatch of Power Systems, In Proceedings of 3rd IEEE International Conference on Power Electronics and Intelligent Transportation System (PEITS 2010), Shenzhen, China, 2010.
[21] J. Kennedy and R. Eberhart. Particle Swarm Optimization, in Proc. IEEE Int. Conf. Neural Networks (ICNN'95), Perth, Australia, 1995, pp. 1942-1948.
[22] Y. Shi and R. Eberhart. A Modified Particle Swarm Optimizer, Proceedings of IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, 1998, pp. 69-73.
[23] Hardiansyah, A Modified Particle Swarm Optimization Technique for Economic Load Dispatch with Valve-Point Effect, International Journal of Intelligent Systems and Applications, 2013, 5(7), pp. 32-41.
[24] Hardiansyah, Junaidi, Yohannes M. S., Solving Economic Load Dispatch Problem Using Particle Swarm Optimization Technique, International Journal of Intelligent Systems and Applications, 2012, 4(12), pp. 12-18.
[25] Hardiansyah, A Novel Hybrid PSO-GSA Method for Non-convex Economic Dispatch Problems, International Journal of Information Engineering and Electronic Business, 2013, 5(5), pp. 1-9.
[26] M. Clerc, J. Kennedy. The Particle Swarm—Explosion, Stability, and Convergence in a Multidimensional Complex Space. IEEE Transactions on Evolutionary Computation, Vol. 6, No. 1, pp. 58-73, 2002.
[27] S. H. Zahiri, Swarm Intelligence and Fuzzy Systems, Nova Publisher, USA, 2010.

[28] Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012). Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems. Computers and Structure. 110(1): 151-166.
[29] Sarvi M, Soltani I, Avanaki IN (2014). A Water Cycle Algorithm Maximum Power Point Tracker for Photovoltaic energy conversion system under Partial Shading Condition. Applied mathematics in Engineering, Management and Technology. 2(1): 103-116.
[30] M. Ashouri, S. M. Hosseini, Application of New Hybrid Particle Swarm Optimization and Gravitational Search Algorithm for Non Convex Economic Load Dispatch Problem, Journal of Advances in Computer Research, 2013, 4(2), pp. 41-51.