Journal of Computers Vol. 33 No. 4, August 2022, pp. 145-158
doi: 10.53106/199115992022083304012

A Power System Profitable Load Dispatch Based on Golden Eagle Optimizer

Jun-Feng Guo[1], Yong-Qiang Zhang[1], Sheng-Bin Xu[1], Trong-The Nguyen[2,3*], Jin-Yang Lin[1]

[1] Research Center for Microelectronics Technology, Fujian University of Technology, Fuzhou 350118, China
{1187349600,1372148500,441052302, 7333744}@qq.com
[2] Multimedia Communications Lab., VNU-HCM, University of Information Technology, Vietnam,
[3] Vietnam National University, Ho Chi Minh City 700000, Vietnam
thent@uit.edu.vn

Received 25 October 2021; Revised 15 January 2022; Accepted 15 February 2022

Abstract. Profitable load dispatch (PLD) is a typical multi-constrained nonlinear optimization problem considered an essential vital part of the power system to achieve energy-saving and consumption reduction. Dealing with the PLD problem using additional methods, e.g., gradient computing quadratic programming, would suffer from computational time complexity. The swarm intelligence optimization algorithm is one of the most promising effective ways of dealing with nonlinear optimization problems like the PLD issue. Golden Eagle optimizer (GEO) is a recent robust swarm intelligence optimization algorithm that has advantages as a few parameters, easy implementation, and powerful search capability. This study suggests a solution to the actual operation constraints of the power system of the PLD model based on novel GEO. The sum of a series of piecewise quadratic polynomials is modeled for the fitness function as the cost function used for figuring optimization out by the first-time GEO. In the experimental section, the IEEE-bus benchmark of 15 and 40-unit test systems are used as the case study to test the performance of the proposed scheme system. The results show that the proposed scheme can solve the power system PLD problem with good robustness and significant economic benefits.

Keywords: profitable load dispatch, Golden Eagle Optimizer, operation constraints, robustness

1 Introduction

The power system's load dispatch determines the output of each power generating unit in a given period [1]. The load demand is provided so that each branch can share the load demand while still meeting the actual limits and reducing the overall system's total operating cost [2]. Profitable load dispatch (PLD) is a critical issue in modern power systems, and it plays a crucial role in improving power system economics [3]. A power system load economic dispatch model considers the upper and lower limit restrictions and power balancing requirements [4]. Many actual nonlinear constraints in the power system operation, such as unit forbidden zone constraints, valve point effects, and ramp rate limitations, are also considered in PLD problems in practical implementations [5]. As a result, the power system's PLD problem is a nonlinear, highly nonlinear multi-constraint optimization problem. The valve point effect complicates the solution of the PLD problem by introducing a large number of local optimums into the process [5].

As the complexity of the PLD problem in power systems with the requirement of profit lead efficiency of power generation cost grows, the optimization methods could have faced complex computation time in solving the PLD problem of economic operation of power system load [1]. Dealing with the PLD problem using additional methods, e.g., gradient computing quadratic programming would suffer from computational time complexity that caused flaws and deficiencies, such as insufficient precision, slow convergence speed, and the tendency to fall into local optimality [6].

The swarm intelligence optimization algorithm is one of the most effective ways to carry out many practical works on solving complicated nonlinear problems [7-8]. The intelligent swarm optimization algorithm is widely employed in various disciplines, including technology, health, society, and finance, and is particularly good at meeting time constraints. In electrical engineering such as the PLD model for power systems in recent years [6], the intelligent swarm optimization method has emerged as one of the most promising optimization techniques for solving severely constrained non-linear and non-convex optimization problems [9]. In modern power system

* Corresponding Author

research, a popular area is applying various algorithms to address profitable dispatching challenges in power systems while enhancing accuracy and practicality [10]. The typical swarm intelligent algorithms can be mentioned as the particle swarm optimization (PSO) [11], differential evolution (DE) algorithm [12], artificial immune algorithm (AIA) [13], and Moth-flame optimization (MFO) [14], widely applied success in computational optimization [15].

Golden Eagle Optimizer (GEO) [16] is a newly proposed swarm intelligence optimization algorithm. The core inspiration of GEO is the intelligence of golden eagles in tuning speed at different stages of their spiral trajectory for hunting. The principle GEO algorithm has several advantages [16], e.g., few parameters and easy to implement, it is potential dealing well with engineering field of power system balancing nonlinear problems like the optimization PLD of a large-scale power system.

This paper suggests a new solution to the large-scale power system PLD problem based on the GEO algorithm [16]. A relatively actual operation comprehensive load profitable dispatch is established as a scheduling model by considering practical constraints, e.g., the ramp rate limit, prohibited operating zone, valve point effect, and network transmission power loss. The power balance constraint is figured out by applying the penalty function for power allocation. The profitable dispatching mathematical model of the power system is tested with IEEE benchmark 15 and 40 unit buses [17].

The innovations of this paper are highlighted as contributions briefly follows.

- For the first time, a recent robust swarm intelligence optimization algorithm based on Golden Eagle optimizer (GEO) is suggested to deal with the large-scale power system PLD problem.
- A relatively actual operation comprehensive load profitable dispatch is established to the objective function as a scheduling model by considering practical constraints, e.g., the ramp rate limit, prohibited operating zone, valve point effect, and network transmission power loss.
- The power balance constraint is figured out by applying the penalty function for power allocation optimized by using the GEO algorithm.
- IEEE benchmark 15 and 40 unit buses are used to verify the suggested applied GEO algorithm for the profitable dispatching mathematical model of the power system.

The rest of the paper is organized as follows. Section 2 reviews models of the PLD problem and GEO algorithm as related work. Section 3 introduces the operation comprehensive load profitable dispatch figured out by applying the GEO algorithm. Section 5 shows and analyses the simulation results. Section 6 concludes the paper.

2 Related Work

This section reviews the PLD model as the power load profitable dispatch problem statement and the original GEO algorithm. The presentation detail is conducted as subsection follows.

2.1 Profitable Load Dispatch Problem

The unit operating cost of economic distribution of load among operating units is generally expressed as a function of unit output. The cost function of the power system is regarded as the sum of a series of quadratic polynomials [18]. Profitable load dispatch (PLD) is modeled mathematically with the cost function of power system load economic dispatch within the upper and lower limit restrictions to balance load power requirements in generation power systems [19]. The objective function of the PLD problem can be expressed as follows.

$$ninF_{cost} = \sum_{i=1}^{M} F_i(P_i) \ , \tag{1}$$

where F_{cost} is the total cost of a power generation system; M is the number of units in the system; P_i is the active power output value of the -th generator. $F_i(P_i)$ represents the generating cost of the i-th generator. The mathematical models of the objective cost function of the energy source power generation of static power systems can be expressed by the power generation cost of the fire motor group its consumption characteristic function.

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

where a_i, b_i and c_i are the cost coefficients of the i-th generator. If the valve point effect is taken into account, the objective function can be expressed as follows.

$$minF_{cost} = \sum_{i=1}^{T}\sum_{i=1}^{M} F_i(P_i) + |e_i sin[z_i(P_{imin} - P_i)]| \ , \tag{3}$$

where e_i and z_i are effect coefficients of the i-th generator with valve points; P_{imin} is the lower limit of the active power output of the i-th generator. The objective function can be operated under constraints, e.g., power balance, generator output, generator ramp rate, and prohibited operating zone constraints.

Power balance constraint-The system power balance constraint is composed of the active power output of the unit, the system network loss, and the total load of the system.

$$\sum_i^M P_i - P_{loss} - P_{load} = 0 \ , \tag{4}$$

where P_{loss} is the system network loss; P_{load} is the total load of the system. The system network loss is obtained by the B coefficient method as shown below.

$$P_{loss} = \sum_{i=1}^{M}\sum_{j=1}^{M} P_i B_{ij} P_{tj} + \sum_{i=1}^{M} B_{oi} P_i + B_{oo} \ , \tag{5}$$

where B_{ij}, B_{oi} and B_{oo} are network loss coefficients.

Generator output constraint is given as.

$$P_{imin} \le P_i \le P_{imax} \ , \tag{6}$$

where P_{imin}, P_{imax} are respectively the lower limit and upper limit of active power output of the i-th generator.

Generator ramp rate constraint is presented as.

$$-DR_i \le P_i - P_{0i} \le UR_i \ , \tag{7}$$

where DR_i, UR_i are the maximum value of output deceleration and maximum value of output growth of unit I respectively. P_{0i} is the active work of the i-th generator at the last moment.

Prohibited Operating Zone constraint- In an operation state, power generating units are some sub-intervals in their operation interval; that is, sub-intervals, the vibration amplitude of the unit bearing will be too large. Therefore, it is necessary to set the operation exclusion zone in the operation interval to avoid these sub-spaces and prevent excessive vibration of the unit bearing. The operating interval of the operation exclusion zone is set as expression as.

$$\begin{cases} P_{imin} \le P_i \le P_i^{dj} \\ P_i^{h(j-1)} \le P_i \le P_i^{dj} \\ P_i^{hN_g} \le P_i \le P_{imax} \end{cases} \ , \tag{8}$$

where P_i^{dj} is the lower limit of the j-th prohibited operating zone of the i-th generator. P_i^{hj} is the upper limit of the j-th prohibited operating zone of the i-th generator. N_g is the total number of prohibited operating zones for the i-th generator.

2.2 Golden Eagle Optimizer Principle

The golden eagle optimizer (GEO) is a recently released swarm intelligent algorithm inspired by the golden eagle and the target prey with essential groups [16]. The golden eagle individual represents the candidate solution of the optimization problem. The prey is the target around the golden eagle, who chooses to cruise until the number of iterations. The prey can be used as a "weathervane" for the golden eagle to proceed in the feasible search space. Each golden eagle cruises around a prey. In the early iteration phase, when the cruising intention is more substantial when a better solution is found, it updates its memory of the optimal prey. Over the iteration, the attack tendency of the golden eagle is stronger [16]. Compute golden eagle's current attack vector is expressed as follows.

$$\overrightarrow{A_1} = \vec{X}_f^* - \vec{X}_i \ , \tag{9}$$

where $\overrightarrow{A_1}$ is the attack vector of *i*-th golden eagle, $\overrightarrow{X_f^*}$ is the best location (prey) golden eagle f has ever visited. $\vec{X}_i$ Is the current location of *i*-th golden eagle. Calculating the cruise vector is tangent hyperplane calculation that is the scalar form in n-dimensional space.

$$h_1 x_1 + h_2 x_2 + \cdots + h_n x_n = d = \sum_{j=1} h_j x_j = d \ , \tag{10}$$

where $\vec{H} = [h_1, h_2, \ldots, h_n]$ is the normal vector; $\vec{X} = [x_1, x_2, \ldots, x_n]$ is the variables vector; $\vec{P} = [p_1, p_2, \ldots, p_n]$ P is random point on the hyperplane; and $d = \vec{H} \cdot \vec{P} = \sum_{i=1}^{n} h_j p_j$. Regarding the position $\vec{X}_i$ as any point on the hyperplane, and taking $\overrightarrow{A_1}$ as the normal of the hyperplane, then we can obtain the hyperplane of $\vec{C}_i^t$ (the cruising vector of *i*-th golden eagle in iteration t) is given as.

$$\sum_{j=1}^{n} a_j x_j = \sum_{j=1}^{n} a_j^t x_j^* \ , \tag{11}$$

Where A_i is set to $\{a_1, a_2, ..., a_n\}$ that is the attack vector; X_i is set to $\{x_1, x_2, ..., x_n\}$ is the decision/design variables vector, $X^* = \{x_1^*, x_2^*, ..., x_n^*\}$ is the location of the selected prey. After calculating the cruise hyperplane of the eagle in the iteration, the cruise vector of the *i*-th golden eagle can be found in this hyperplane as follows.

$$c_k = \frac{d - \sum_{j,j \neq k} a_j}{a_k} \ , \tag{12}$$

where c_k is the *k*-th element of the destination point C. a_j is the *j*-th element of the attack vector A_i. *d* is the right-hand side of the. (10). a_k^t is the *k*-th element of the attack vector A_i, and *k* is the index of the fixed variable. Random target points on the cruising hyperplane is then the general representation of the target point on the cruise hyperplane.

$$\vec{C}_i = \left(c_1 = random, c_2 = random, \ldots, c_k = \frac{d -, \sum_{j,j \neq k} a_j}{a_k}, \ldots, c_n = random \right), \tag{13}$$

The eagle's displacement is measured by attack and vector. The step size vector of the golden eagle is defined in iteration as follows.

$$\Delta x_i = \vec{r}_1 p_a \frac{\vec{A}_i}{\|\vec{A}_i\|} + \vec{r}_2 p_c \frac{\vec{C}_i}{\|\vec{C}_i\|} \ , \tag{14}$$

where p_a^t is the attack coefficient in iteration t. p_c^t is the cruise coefficient in iteration t. $\vec{r}_1$ and $\vec{r}_2$ are random vectors whose elements lie in the interval [0,1]. The calculation of the Euclidean norm of $\|\vec{A}_i\|$ and $\|\vec{C}_i\|$ is expressed as follows.

$$\|\vec{A}_i\| = \sqrt{\sum_{j=1}^{n} a_j^2} \ , \ \|\vec{C}_i\| = \sqrt{\sum_{j=1}^{n} c_j^2} \ , \tag{15}$$

The golden eagle position update formula is shown following expression.

$$x^{t+1} = x^t + \Delta x_i^t , \tag{16}$$

If the fitness of the golden eagle's new position is better than its remembered position, the eagle's memory is updated to the new place. Otherwise, the memory location remains the same, but the eagle will reside in the new location. In the latest iteration, each golden eagle randomly selects a golden eagle from the population to hover around the position with the best memory, calculates the attack vector, calculates the cruise vector, and finally calculates the step vector and the new position of the next iteration. The loop is executed until any termination conditions reach. Tow parameters p_a and p_c are used to shift from exploration to exploitation in GEO [16]. In the initial stage, p_a is low and p_c is high that is expressed as iteration progresses, p_a gradually increases and p_c gradually decreases.

$$\begin{cases} p_a = p_a^0 + \frac{t}{T}|p_a^T - p_a^0| \\ p_c = p_c^0 + \frac{t}{T}|p_c^T - p_c^0| \end{cases}, \tag{17}$$

Where t represents the current number of iterations, and T represents the maximum number of iterations. p_a^0 and p_a^T are the initial and final values of the attack propensity respectively. p_c^0 and p_c^T are the initial and final values of cruise tendency respectively. $[p_a^0,\ p_a^T]$ is set to [0.5,2] and $[p_c^0,\ p_c^T]$ is set to [1,0.5] that means that p_a; p_c linearly increases from 0.5 and 0.5 to the end of the iteration.

3 A Solution to PLD Based on Golden Eagle Optimization

The PLD problem is multi-constrained, nonlinear, discrete, and other characteristics in the power system include equality and inequality constraints. Obtained solutions for individuals must satisfy the constraints in the solution space. The objective function value of each individual needs to be processed to meet all constraints [20]. The optimization variable of the PLD problem of the power system is the active power output of each unit, and the dimension of the problem is equal to the sum of the number of units. Combined with the GEO algorithm and PLD model characteristics, each golden eagle represents a candidate solution to the power system's PLD problem. Each golden eagle's position is a vector composed of the output of each unit.

$$X = \begin{bmatrix} x_1^1 & \dots & x_i^1 & \dots & x_n^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^i & \dots & x_i^i & \dots & x_n^i \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^n & \dots & x_i^n & \dots & x_n^n \end{bmatrix}, \tag{18}$$

where X is the output golden eagle matrix of the generator set, and the row vectors of matrix X represent the specific positions of each eagle. Unit output constraint processing is figured out with as the golden Eagle optimization algorithm needs to randomly generate the golden eagle position in the preliminary trial stage, as shown in formula (19):

$$X = L_{\text{bound}} + R_{rand}(U_{\text{bound}} - L_{\text{bound}}), \tag{19}$$

where R_{rand} is a random number evenly distributed between 0 and 1. L_{bound} is the output lower limit matrix of the generator set and U_{bound} is the output upper limit matrix of the generator set.

In order to satisfy the constraint conditions, the upper and lower limits of unit output are restricted to the output range of each unit by the random position generation formula in the initialization stage in the form of the matrix. A system power's constraint processing is expressed as limiting the updating range space for the golden eagle's position by adding a penalty function for the objective function's constraints that is adopted is rewritten as follows.

$$F = \sum_{i=1}^{T}\sum_{i=1}^{M} F_i(P_{ti}) + |e_i sin[f_i(P_{imin} - P_i)]| + \mu\left(\sum_{i=1}^{M} P_{ti} - P_{tloss} - P_{tload}\right)^2, \tag{20}$$

where F is the reconstructed objective function after the penalty function is added. μ is the penalty coefficient. The golden eagle's memory is updated to the new position if the fitness of the new location is better than the remembered position. The eagle will live at the new site, but the memory location will remain unchanged. In the most recent iteration, each golden eagle chooses a golden eagle randomly from the population to hover over the place with the best memory to calculate the attack vector, cruise vector, step vector, and new position for the next iteration. The loop is executed run until it reaches any termination circumstances. In GEO, the two parameters p_a and p_c are employed to transition from exploration to exploitation. Initially, p_a is low, and p_c is large, as p_a progressively increases, and p_c gradually drops as iteration goes.

The algorithm implementation process is implemented process as follows.

Step 1: Initialize golden Eagle basic parameters and PLD unit parameters. All golden eagles are randomly initialized between the upper and lower bounds of the output capacity constraint of the unit, and the fitness function value is used by Eq (20).

Step 2: Update the position, attack factor, cruise factor, and fitness value of the golden eagle. If it is in the first generation, the current position of the initial population of the golden eagle is directly used as the optimal position of the golden eagle memory. In the iterative process, the golden eagle randomly selects a prey from the population's memory and calculates the current target's corresponding attack coefficient and cruise coefficient.

Step 3: Golden Eagle updates the current position and calculates the fitness value for the new place. Judge whether the fitness value is better than the fitness value in the memory of the golden eagle; if it is better than the fitness value in the memory of the golden eagle, update the optimal position in the memory of the golden eagle, and otherwise do not change.

Step 4: Termination condition: The iteration is terminated if the maximum number of iterations is reached. Output the optimal dispatching scheme (the best golden eagle memory) and power generation cost (the best fitness value of golden eagle) for the PLD problem; otherwise, go to Step 2 and repeat the iterative process until the end met termination condition.

4. Simulation and Discussion Results

Two power generation systems of IEEE benchmark with the unit number are 15 and 40, respectively, are used to verify the reliability and effectiveness of the proposed method. Unit 15 and 40 test systems are static systems with 10500MW and 2630MW of total load demand, respectively, not considering or ignoring the embargo zone, unit climbing, and the network loss constraints [17]. The unit test system also considers the slow variable rate, and each unit weighs the upper and lower output limits, embargoed zone constraints, and network loss.

The results of the GEO [16] are compared with the MFO [21], PSO [22], and gray GWO [23]. The number of search agents is uniformly set to 30 in all algorithms during the simulation, and the maximum number of iterations is 1000. Each instance is run separately multiple times to ensure the tests' effectiveness, comparability, and robustness. For example, the parameters of setting for 15 machines test system, e.g., system prohibited zones of generating units, the total load demand, and the coefficients B: [B_{ij}, B_{oi}, B_{oo}] are listed in Table 1 and power factors.

Table 1. The setting for a test system prohibited zones of generating units in an experiment

Unit	Prohibited zones (MW)
2	[185 225] [305 335] [420 450]
5	[180 200] [305 335] [390 420]
6	[230 255] [365 395] [430 455]
12	[30 40] [55 65]

The power loss factor B is expressed as follows.

$$B_{ij} = \begin{bmatrix} 0.0014 & 0.0012 & 0.0007 & -0.0001 & -0.0003 & -0.0001 & -0.0001 & -0.0001 & -0.0003 & 0.0005 & -0.0003 & -0.0002 & 0.0004 & 0.0003 & -0.0001 \\ 0.0012 & 0.0015 & 0.0013 & 0.0000 & -0.0005 & -0.0002 & 0.0000 & 0.0001 & -0.0002 & -0.0004 & -0.0004 & 0.0000 & 0.0004 & 0.0010 & -0.0002 \\ 0.0007 & 0.0013 & 0.0076 & -0.0001 & -0.0013 & -0.0009 & -0.0001 & 0.0000 & -0.0008 & -0.0012 & -0.0017 & 0.0000 & -0.0026 & 0.0111 & -0.0028 \\ -0.0001 & 0.0000 & -0.0001 & 0.0034 & -0.0007 & -0.0004 & 0.0011 & 0.0050 & 0.0029 & 0.0032 & -0.0011 & 0.0000 & 0.0001 & 0.0001 & -0.0026 \\ -0.0003 & -0.0005 & -0.0013 & -0.0007 & 0.0090 & 0.0014 & -0.0003 & -0.0012 & -0.0010 & -0.0013 & 0.0007 & -0.0002 & -0.0002 & -0.0024 & -0.0003 \\ -0.0001 & -0.0002 & -0.0009 & -0.0004 & 0.0014 & 0.0016 & 0.0000 & -0.0006 & -0.0005 & -0.0008 & 0.0011 & -0.0001 & -0.0002 & -0.0017 & 0.0003 \\ -0.0001 & 0.0000 & -0.0001 & 0.0011 & -0.0003 & 0.0000 & 0.0015 & 0.0017 & 0.0015 & 0.0009 & -0.0005 & 0.0007 & 0.0000 & -0.0002 & -0.0008 \\ -0.0001 & 0.0001 & 0.0000 & 0.0050 & -0.0012 & -0.0006 & 0.0017 & 0.0168 & 0.0082 & 0.0079 & -0.0023 & -0.0036 & 0.0001 & 0.0005 & -0.0078 \\ -0.0003 & -0.0002 & -0.0008 & 0.0029 & -0.0010 & -0.0005 & 0.0015 & 0.0082 & 0.0129 & 0.0116 & -0.0021 & -0.0025 & 0.0007 & -0.0012 & -0.0072 \\ -0.0005 & -0.0004 & -0.0012 & 0.0032 & -0.0013 & -0.0008 & 0.0009 & 0.0079 & 0.0116 & 0.0200 & -0.0027 & -0.0034 & 0.0009 & -0.0011 & -0.0088 \\ -0.0003 & -0.0004 & -0.0017 & -0.0011 & 0.0007 & 0.0011 & -0.0005 & -0.0023 & -0.0021 & -0.0027 & 0.0140 & 0.0001 & 0.0004 & -0.0038 & 0.0168 \\ -0.0002 & 0.0000 & 0.0000 & 0.0000 & -0.0002 & -0.0001 & 0.0007 & -0.0036 & -0.0025 & -0.0034 & 0.0001 & 0.0054 & -0.0001 & -0.0004 & 0.0028 \\ 0.0004 & 0.0004 & 0.0001 & 0.0001 & -0.0002 & -0.0002 & 0.0000 & 0.0001 & 0.0007 & 0.0009 & 0.0004 & -0.0001 & 0.0103 & -0.0101 & 0.0028 \\ 0.0003 & 0.0010 & 0.0001 & 0.0001 & -0.0024 & -0.0017 & -0.0002 & 0.0005 & -0.0012 & -0.0011 & -0.0038 & -0.0004 & -0.0101 & 0.0578 & -0.0094 \\ -0.0001 & -0.0002 & -0.0026 & -0.0026 & -0.0003 & 0.0003 & -0.0008 & -0.0078 & -0.0072 & -0.0088 & 0.0168 & 0.0028 & 0.0028 & -0.0094 & 0.1283 \end{bmatrix}$$

$$B_{0i} = [-0.0001 \quad -0.0002 \quad 0.0028 \quad -0.0001 \quad 0.0001 \quad -0.0003 \quad -0.0002 \quad -0.0002 \quad 0.0006 \quad 0.0039 \quad -00017 \quad 0.0000 \quad -0.0032 \quad 0.0067 \quad -0.0064]$$

$$B_{oo} = 0.0055$$

Table 2. The selected 15-unit test system parameters

Units	α ($/MW^2)	β ($/MW)	γ ($)	UR (MW/h)	*LR* (MW/h)	P_0	P_{min} (MW)	P_{max} (MW)
1	0.000299	10.1	671	80	120	400	150	455
2	0.000183	10.2	574	80	120	300	150	455
3	0.001126	8.8	374	130	130	105	20	130
4	0.001126	8.8	374	130	130	100	20	130
5	0.000205	10.4	461	80	120	90	150	470
6	0.000301	10.1	630	80	120	400	135	460
7	0.000364	9.8	548	80	120	350	135	465
8	0.000338	11.2	227	65	100	95	60	300
9	0.000807	11.2	173	60	100	105	25	162
10	0.001203	10.7	175	60	100	110	25	160
11	0.003586	10.2	186	80	80	60	20	80
12	0.005513	9.9	230	80	80	40	20	80
13	0.000371	13.1	225	80	80	30	25	85
14	0.001929	12.1	309	55	55	20	15	55
15	0.004447	12.4	323	55	55	20	15	55

Table 2 and Table 3 show the machine parameters of test systems of 15 and 40 power generating plant units with the total load total demands, respectively.

Table 3. The selected 40-unit test system parameters

Units	*A* ($/ MW^2)	β ($/*MW*)	r ($)	e	f	P_{min} (MW)	P_{max} (MW)
1	0.00690	6.73	94.705	100	0.084	36	114
2	0.00690	6.73	94.705	100	0.084	36	114

3	0.02028	7.07	309.54	100	0.084	60	120
4	0.00942	8.18	369.03	150	0.063	80	190
5	0.0014	5.35	148.89	120	0.077	47	97
6	0.00142	8.05	222.33	100	0.084	68	140
7	0.00357	8.03	287.71	200	0.042	110	300
8	0.00492	6.99	391.98	200	0.042	135	300
9	0.00573	6.60	455.76	200	0.042	135	300
10	0.00605	12.9	722.82	200	0.042	130	300
11	0.00515	12.9	635.20	200	0.042	94	375
12	0.00569	12.8	654.69	200	0.042	94	375
13	0.00421	12.5	913.40	300	0.035	125	500
14	0.00752	8.84	1760.4	300	0.035	125	500
15	0.00708	9.15	1728.3	300	0.035	125	500
16	0.00708	9.15	1728.3	300	0.035	125	500
17	0.00313	7.97	647.85	300	0.035	220	500
18	0.00313	7.95	649.69	300	0.035	220	500
19	0.00313	7.97	647.83	300	0.035	242	550
20	0.00313	7.97	647.81	300	0.035	242	550
21	0.00298	6.63	785.96	300	0.035	254	550
22	0.00298	6.63	785.96	300	0.035	254	550
23	0.00294	6.66	794.53	300	0.035	254	550
24	0.00294	6.66	794.53	300	0.035	254	550
25	0.00277	7.10	801.32	300	0.035	254	550
26	0.00277	7.10	801.32	300	0.035	254	550
27	0.52124	3.33	1055.1	120	0.077	10	150
28	0.52124	3.33	1055.1	120	0.077	10	150
29	0.52124	3.33	1055.1	120	0.077	10	150
30	0.01140	5.35	148.89	120	0.077	47	97
31	0.00160	6.43	222.92	150	0.063	60	190
32	0.00160	6.43	222.92	150	0.063	60	190
33	0.00160	6.43	222.92	150	0.063	60	190
34	0.00010	8.95	107.87	200	0.042	90	200
35	0.00010	8.62	116.58	200	0.042	90	200
36	0.00010	8.62	116.58	200	0.042	90	200
37	0.01610	5.88	307.45	80	0.098	25	110
38	0.01610	5.88	307.45	80	0.098	25	110
39	0.01610	5.88	307.45	80	0.098	25	110
40	0.00313	7.97	647.83	300	0.035	242	550

Fig. 1 shows the suggested scheme's obtained graph of the daily power load loss distribution for the test system of 15 units. Because the load consumption is highest hours between 9 a.m. and 12 p.m., the power load loss is likewise significant, resulting in the yellow color indicated in Fig 1.

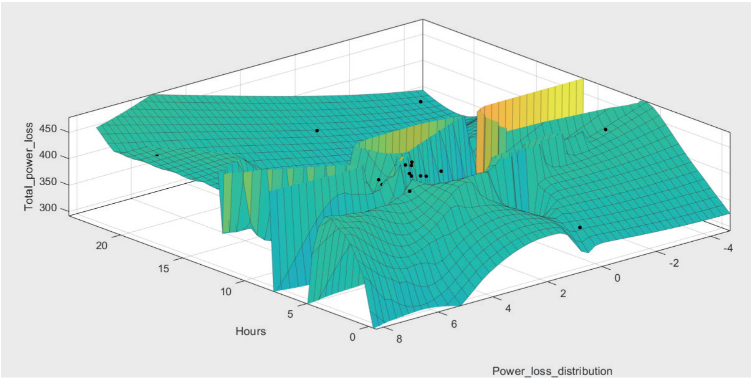


Fig. 1. The suggested scheme obtained graph of the daily power load loss distribution for the test system of 15 units

Fig. 2 displays the comparison of iteration times in convergence of the GEO with the other methods, PSO, FMO, and GWO, under the case of a 15 unit system. It is seen that the GEO produces the convergence speed faster than the other comparison methods.

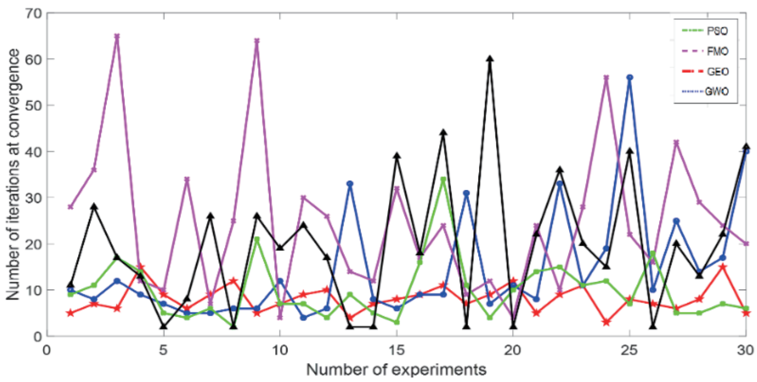


Fig. 2. Comparison of iteration times in convergence under the case of a 15 unit system

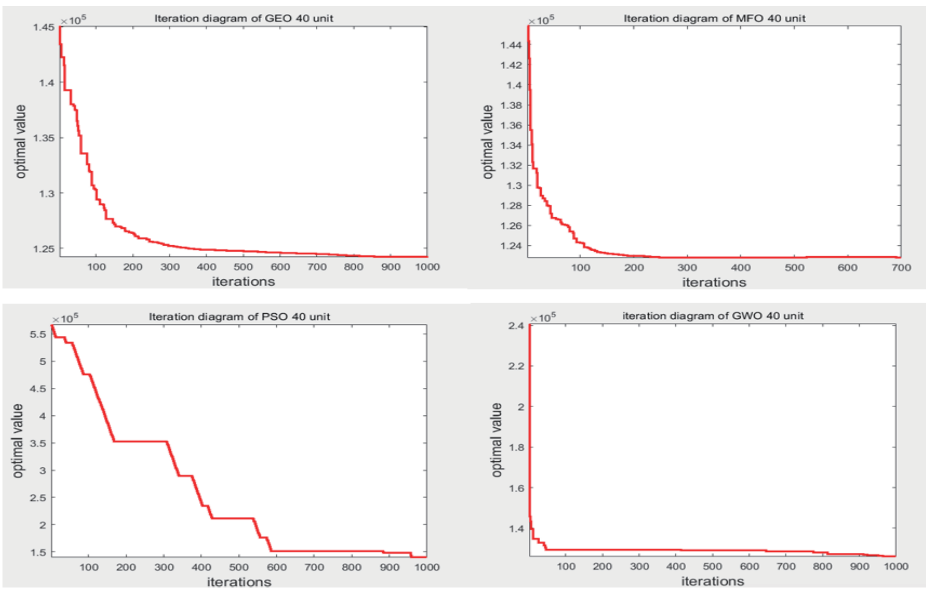
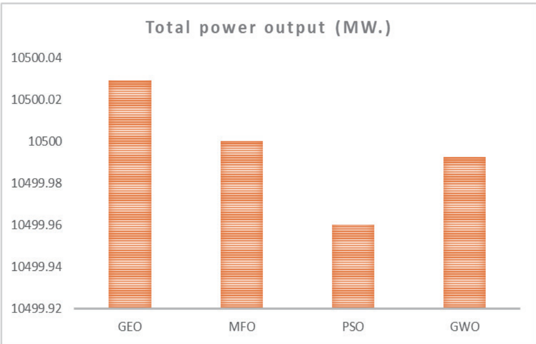


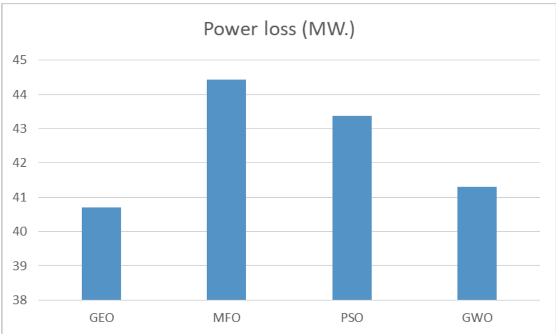
Fig. 3. The calculating iteration diagram of the algorithms for the test systems of 40 units

Fig. 3 shows the comparison results curves of the quality performance in terms of convergence speed and time consumption of the GEO optimization method with PSO, FMO, and GWO algorithms. The observation results show that the GEO is superior to PSO and GWO methods when calculating the test systems and finding the optimal target value.

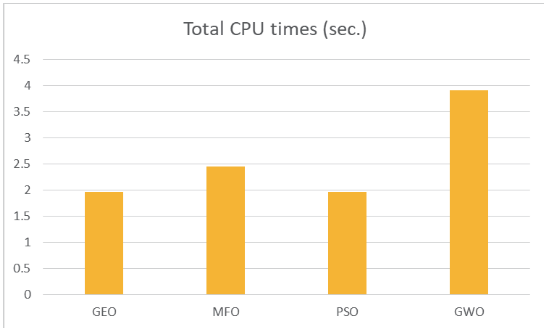


(a) Total power output (MW.)

(b) Total generation cost ($/h.)



(c) Total power loss (MW.)



(d) Total CPU times (sec.)

Fig. 4. The comparison obtained result comeouts of the four algorithms e.g., the GEO, MFO, PSO and GWO for the test systems of 15 units

Fig. 4 and Fig. 5 compare the suggested method with MFO, PSO, and GWO for the power test systems of 15 and 40 units, respectively, regarding the (a) total power outcomes, (b) consuming power cost, (c) power load loss, and (d) executing run times.

The suggested method produces the highest bar number in subfigures (a) for total power output, and the executing run time (c) of the GEO is as short as the PSO. Moreover, the suggested scheme has obtained figures more minor than the other MFO, PSO, and GWO for the generating power cost and the power load loss.

Table 4. Comparison of results of different algorithms (15-unit test system)

Units output	**GEO**	MFO	PSO	GWO
P_1	424.88380	383.09330	445.81340	358.04970
P_2	413.97400	455	411.62730	376.27940
P_3	110.38160	130	128.01580	116.20430
P_4	130	20	121.88440	69.09725
P_5	364.00920	259.02820	248.62000	438.59490
P_6	335.44610	460	305.74030	335.62150
P_7	338.09530	465	202.33120	376.97400
P_8	171.00300	60	220.80620	144.09410
P_9	122.86760	148.01680	152.45640	73.77481
P_{10}	89.84261	42.97372	154.58030	142.02190
P_{11}	53.68711	73.95649	72.45328	61.41507

P_{12}	26.18889	45.37251	52.15760	46.55478
P_{13}	31.39906	25	74.27055	65.57457
P_{14}	29.21749	37.00203	41.06329	36.35418
P_{15}	29.69113	52.98635	44.51898	29.66419
Total power output (MW.)	2670.6868	2657.4294	2676.3390	2670.2747
Total generation cost ($/h.)	**33571.960**	33778.830	33874.398	33854.010
Power loss (MW.)	40.7131	47.4294	46.3694	41.2985
Deviation	0.026237	0.02873	0.030126	0.023852
Total CPU times (sec.)	1.212698	1.231029	1.213919	1.629856

Table 5. Comparison of results of different algorithms (40-unit test system)

Units output	**GEO**	MFO	PSO	GWO
P_1	111.40940	114	103.38030	70.73795
P_2	110.65340	114	73.27366	69.12132
P_3	97.43179	60	105.84260	118.63920
P_4	179.93980	190	135.55590	135.98520
P_5	91.61706	97	72.92917	75.43688
P_6	121.23800	140	101.57150	88.81333
P_7	264.08250	300	236.38130	296.88970
P_8	284.52580	300	250.62590	294.37540
P_9	286.03790	300	263.16830	297.05950
P_{10}	204.85560	130	280.49500	202.14850
P_{11}	245.87230	94	308.29600	245.19510
P_{12}	243.27080	375	248.17710	210.42150
P_{13}	304.49720	125	358.47850	128.36970
P_{14}	394.32000	214.7598	461.76500	389.42230
P_{15}	304.51930	304.5196	397.40950	474.59730
P_{16}	304.20510	304.5196	445.64960	391.84470
P_{17}	489.35320	500	482.57070	499.85940
P_{18}	449.63330	500	466.65340	494.42030
P_{19}	423.72540	550	457.42620	523.38490
P_{20}	498.79550	421.5196	518.29110	531.59490
P_{21}	523.34330	523.2794	488.27000	524.97290
P_{22}	523.46480	523.2794	522.43480	523.59310
P_{23}	523.21880	550	460.77470	525.97740
P_{24}	523.54380	523.2794	461.41460	527.41550
P_{25}	523.52630	550	364.93100	524.89060
P_{26}	523.35530	523.2794	513.43200	534.87650
P_{27}	10.21827	10	109.88220	15.13035
P_{28}	10.00001	10	46.86652	32.58896
P_{29}	10.0001	10	90.53189	11.62350
P_{30}	88.74756	92.56397	62.82164	84.25475
P_{31}	177.40290	190	171.27360	168.82260
P_{32}	165.92760	190	180.08850	166.49010
P_{33}	166.34700	190	181.90830	182.78720
P_{34}	164.84170	200	134.36170	181.81200
P_{35}	174.27570	200	139.38390	94.96204
P_{36}	178.67560	200	196.25660	164.69250
P_{37}	109.99200	110	70.84436	26.72955
P_{38}	89.28232	110	45.42183	35.59133

P_{39}	92.57484	110	63.41170	110
P_{40}	511.3085	550	427.70480	524.46460
Total power output (MW.)	10500.0293	10500.0002	10499.96	10499.9926
Total generation cost ($/h.)	**122928.8**	123249.7	139875.5	125994.7
Power loss (MW.)	60.7131	64.4294	63.3694	61.2985
Standard Deviation	0.0293	0.0302	0.04448	0.02736
Total CPU times (sec.)	1.957508	2.451893	1.962717	3.908980

Table 4 and Table 5 show the comparison of the suggested method with MFO, PSO, and GWO for test systems of 15 and 40 units, respectively; where the P_1, P_2, ..., P_n are the generator output power of each branch, n is a number of solution units. It can be seen that the proposed scheme produces a better performance of optimization costs than the other comparison methods. An iterative diagram of solving the system scheduling problem of 15 units based on the GEO and different algorithms, e.g., the MFO, PSO, GWO algorithms for the PLD under the same conditions.

(a) Total power output (MW.)

(b) Total generation cost ($/h.)

(c) Total power loss (MW.)

(d) Total CPU times (sec.)

Fig. 5. The comparison obtained result comeouts of the four algorithms e.g., the GEO, MFO, PSO and GWO for the test systems of 40 units

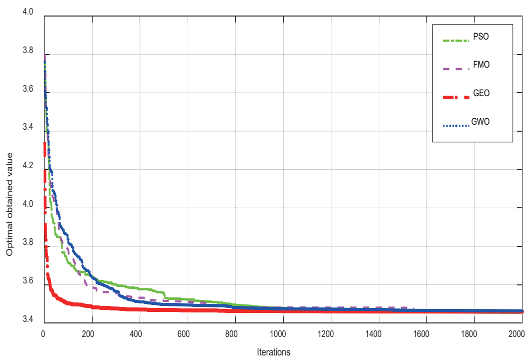


Fig. 6. The comparison obtained result curves of the suggested algorithm with the other algorithms for the test systems of 40 units

Fig. 6 shows the comparison obtained result curves of the suggested algorithm with the other algorithms. e.g., FMO, PSO, and GWO for the test systems of 40 units. The observed figure shows that the GEO optimization method has better quality performance in convergence speed and time consumption than PSO and GWO methods. In general, we can say that the GEO can solve the PLD problem in the power system with good robustness and significant economic benefits.

5 Conclusion

This paper suggested a new solution for the profitable load dispatch (PLD) allocation problem in the power generating plant system based on the golden Eagle optimization algorithm (GEO). The Golden Eagle optimizer (GEO) is applied for the first time to deal with the PLD problem features. GEO is a new robust swarm intelligence optimization algorithm that owns advantages as a few parameters, easy implementation, exploration and exploitation balance, and powerful search capability. The economic load scheduling allocation with piecewise quadratic polynomials is used to model as the objective function for a typical multi-constrained nonlinear optimization problem that is considered an essential part of the power system to achieve energy-saving and consumption reduction. The objective function was optimized as the PLD model by the GEO. In the experimental part, the performance of the proposed scheme was verified with aspects of economy, rapidity, convergence, and robustness by two kinds of IEEE testing systems, e.g., the fifteen and forty plants. The compared results show that the suggested method can effectively solve the PLD problem, utilizing the GEO algorithm's advantages in terms of fast convergence, good robustness, and easy implementation. In the future, we will implement the suggested scheme with further complex scheduling and scaling the PLD problems by applying the methods of Improved Moth-flame optimization (IMFO) [24], Improved Flower pollination algorithm (IFPA) [25].

Acknowledgments

This work is supported by the National Display Joint Engineering Laboratory for Flat Panel Display Technology (No. KF1802, GY-Z18038), and artially supported by the VNUHCM-University of Information Technology's Scientific Research Support Fund.

References

[1] A. Mahor, V. Prasad, S. Rangnekar, Economic dispatch using particle swarm optimization: A review, Renewable and Sustainable Energy Reviews 13(8)(2009) 2134-2141.
[2] W.T. Elsayed, Y.G. Hegazy, F.M. Bendary, M.S. El-Bages, A review on accuracy issues related to solving the non-convex economic dispatch problem, Electric Power Systems Research 141(2016) 325-332.
[3] L. Jebaraj, C. Venkatesan, I. Soubache, C.C.A. Rajan, Application of differential evolution algorithm in static and dynamic economic or emission dispatch problem: A review, Renewable and Sustainable Energy Reviews 77(2017) 1206-1220.
[4] T.-T. Nguyen, H.-J. Wang, T.-K. Dao, J.-S. Pan, J.-H. Liu, S.-W. Weng, An Improved Slime Mold Algorithm and Its Application for Optimal Operation of Cascade Hydropower Stations, IEEE Access 8(2020) 226754-226772.
[5] 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.
[6] D. He, F. Wang, Z. Mao, Hybrid genetic algorithm for economic dispatch with valve-point effect, Electric Power Systems Research 78(4)(2008) 626-633.
[7] Q.-W. Zhang, F.F. Zhan, A Particle swarm optimization algorithm based on multi-subgroup harmony search, Journal of Computers 31(6)(2020) 116-126.
[8] Y. Yang, X. Zuo, M. Fu, S. Yu, C. Jia, Adaptive Cuckoo Search Algorithm Based on Dynamic Adjustment Mechanism, Journal of Computers 32(5)(2021) 171-183.
[9] M. Gholamghasemi, E. Akbari, M.B. Asadpoor, M. Ghasemi, A new solution to the non-convex economic load dispatch problems using phasor particle swarm optimization, Applied Soft Computing 79(2019) 111-124.
[10]S. Murugan, B. Horák, A review of micro combined heat and power systems for residential applications, Renewable and Sustainable Energy Reviews 64(2016) 144-162.
[11]Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, in: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99, 1999.
[12]J. Pan, N. Liu, S. Chu, A Hybrid Differential Evolution Algorithm and Its Application in Unmanned Combat Aerial Vehicle Path Planning, IEEE Access 8(2020) 17691-17712.
[13]D. Dasgupta, Advances in artificial immune systems, IEEE Computational Intelligence Magazine 1(4)(2006) 40-49.
[14]S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowledge-Based Systems

89(2015) 228-249.
[15]S. Deb, D.S. Abdelminaam, M. Said, E.H. Houssein, Recent methodology-based gradient-based optimizer for economic load dispatch problem, IEEE Access 9 (2021) 44322-44338.
[16]A. Mohammadi-Balani, M.D. Nayeri, A. Azar, M. Taghizadeh-Yazdi, Golden eagle optimizer: A nature-inspired meta-heuristic algorithm, Computers & Industrial Engineering 152(2021) 107050.
[17]P. Kotsampopoulos, D. Lagos, N. Hatziargyriou, M.O. Faruque, G. Lauss, O. Nzimako, P. Forsyth, M. Steurer, F. Ponci, A. Monti, A benchmark system for hardware-in-the-loop testing of distributed energy resources, IEEE Power and Energy Technology Systems Journal 5(3)(2018) 94-103.
[18]T.-T. Nguyen, M.J. Wang, J.S. Pan, T.K. Dao, T.G. Ngo, A Load Economic Dispatch Based on Ion Motion Optimization Algorithm, in: Proc. Advances in Intelligent Information Hiding and Multimedia Signal Processing, 2020.
[19]T.K. Dao, T.S. Pan, T.-T. Nguyen, S.C. Chu, Evolved Bat Algorithm for solving the Economic Load Dispatch problem, in: Proc. Genetic and Evolutionary Computing, 2015.
[20]T.K. Panigrahi, A.K. Sahoo, A. Behera, A review on application of various heuristic techniques to combined economic and emission dispatch in a modern power system scenario, Energy Procedia 138(2017) 458-463.
[21]I.N. Trivedi, A. Kumar, A.H. Ranpariya, P. Jangir, Economic Load Dispatch problem with ramp rate limits and prohibited operating zones solve using Levy flight Moth-Flame optimizer, in: 2016 International Conference on Energy Efficient Technologies for Sustainability (ICEETS), 2016.
[22]A. Zaraki, M.F.B. Othman, Implementing particle swarm optimization to solve economic load dispatch problem, in: 2009 International Conference of Soft Computing and Pattern Recognition, 2009.
[23]V.K. Kamboj, S.K. Bath, J.S. Dhillon, Solution of non-convex economic load dispatch problem using Grey Wolf Optimizer, Neural Computing and Applications 27(5)(2016) 1301-1316.
[24]T.-T. Nguyen, H.-J. Wang, T.-K. Dao, J.-S. Pan, T.-G. Ngo, J. Yu, A Scheme of Color Image Multithreshold Segmentation Based on Improved Moth-Flame Algorithm, IEEE Access 8(2020) 174142-174159.
[25]T.-T. Nguyen, J.S. Pan, T.K. Dao, An Improved Flower Pollination Algorithm for Optimizing Layouts of Nodes in Wireless Sensor Network, IEEE Access 7(2019) 75985-75998.