Abstract—This paper presents a new approach to economic load dispatch (ELD) problems with non-smooth objective functions using a particle swarm optimization (PSO). In practice, ELD problems have non-smooth objective functions with equality and inequality constraints that make it difficult to find the global optimum using any mathematical approaches. In this paper, a new PSO framework is suggested to deal with the equality and inequality constraints in ELD problems. The proposed PSO can always provide solution(s) satisfying the constraints within a realistic computation time and is devised not to interrupt the dynamic process inherent in the conventional PSO. To show its efficiency and effectiveness, the proposed PSO is applied to sample ELD problems with smooth cost functions as well as with non-smooth cost functions. The results of the proposed PSO are compared with those of the conventional numerical method, evolutionary programming approach, and the modified Hopfield neural network approach.

Index Terms—Economic load dispatch, particle swarm optimization, constrained optimization, non-smooth optimization.

I. INTRODUCTION

Most of power system optimization problems including economic load dispatch (ELD) have complex and nonlinear characteristics with heavy equality and inequality constraints. To solve these problems, various salient mathematical approaches have been suggested for the past decades. As an alternative to the conventional mathematical approaches, the heuristic optimization techniques such as genetic algorithms, Tabu search, simulated annealing, and recently-introduced particle swarm optimization (PSO) are considered as realistic and powerful solution schemes to obtain the global or quasi-global optimums in power system optimization problems [1].

Recently, Eberhart and Kennedy suggested a particle swarm optimization (PSO) based on the analogy of swarm of bird and school of fish [2]. The PSO mimics the behaviors of individuals in a swarm to maximize the survival of the species. In PSO, each individual decides his decision using his own experience as well as other individuals’ experiences [3]. The algorithm, which is based on a metaphor of social interaction, searches a space by adjusting the trajectories of moving points in a multidimensional space. The individual particles are drawn stochastically toward the position of present velocity of each individual, their own previous best performance, and the best previous performance of their neighbors [4]. The main advantages of the PSO algorithm are summarized as: simple concept, easy implementation, robustness to control parameters, and computational efficiency when compared with mathematical algorithm and other heuristic optimization techniques.

Recently, PSO have been successfully applied to various fields of power system optimization such as power system stabilizer design [5], reactive power and voltage control [3], and dynamic security border identification [6], etc. The original PSO mechanism is directly applicable to the problems with continuous domain and without any constraints. Therefore, it is necessary to revise the original PSO to reflect the equality/inequality constraints of the variables in the process of modifying each individual’s search.

Yoshida et al. [3] suggested a modified PSO to control reactive power and voltage considering voltage security assessment. Since the problem was a mixed-integer nonlinear optimization problem with inequality constraints, they applied the classical penalty method to reflect the constraint-violating variables. Abido [5] developed a revised PSO for determining the optimal values of parameters for power system stabilizers. In the study, the velocity of each parameter is limited to a certain value to reflect the inequality constraint problem in the dynamic process.

In practice, an ELD problem is represented as a non-smooth optimization problem with equality and inequality constraints, which makes it difficult to obtain the global optimum. To solve the problems, many salient methods have been proposed such as a mathematical approach [7], dynamic programming [8], improved evolutionary programming [9], neural network approaches [10], [11], and genetic algorithm [12].

In this paper, we propose an alternative approach to the non-smooth ELD problems using a new PSO focused on the treatment of the equality and inequality constraints in the process of modifying each individual’s search. The equality constraint (i.e., the supply/demand balance) is easily satisfied by specifying a variable (i.e., generator) at random in each iteration as a slag generator whose value is determined by the
difference between the total system demand and the total generation excluding the slag generator. The inequality constraints in creating initial individuals are easily handled. However, the next position of an individual produced by the PSO algorithm can violate the inequality constraint. In this case, the position of any individual violating the constraints is set to its minimum or maximum position depending on the velocity evaluated.

The feasibility of the proposed PSO for ELD problems with quadratic and piecewise quadratic cost functions is demonstrated and compared with the existing approaches [7]-[11].

II. FORMULATION OF ELD PROBLEM

The objective of an ELD problem is to find the optimal combination of power generations that minimizes the total generation cost while satisfying an equality constraint and inequality constraints. The most simplified cost function of each generator can be represented as a quadratic function. Therefore, the objective function can be described as follows, whose solution can be obtained by the conventional mathematical methods [8]:

\[ C = \sum_{i \in I} a_i + b_i P_i + c_i P_i^2 \]  

where,
- \( C \) : total generation cost,
- \( a_i, b_i, c_i \) : cost coefficients of generator \( i \),
- \( P_i \) : electrical output of generator \( i \),
- \( I \) : set for all generators.

While minimizing the total generation cost, the following constraints should be satisfied. For supply/demand balance, the total generated power should be equal to the total system demand plus the transmission network loss. However, the network loss is not considered in this paper for simplicity. Therefore, the equality constraint is expressed as follows:

\[ \sum_{i \in I} P_i = D \]  

where \( D \) is the total system demand.

Also, generation output of each unit should be between its maximum and minimum limits. That is, the following inequality constraint for each generator should be satisfied.

\[ P_{i,\text{min}} \leq P_i \leq P_{i,\text{max}} \]  

where,
- \( P_{i,\text{min}}, P_{i,\text{max}} \) : minimum, maximum output of generator \( i \).

In reality, the objective function of an ELD problem has discontinuous and nondifferentiable points according to valve loading and change of fuels, so the objective function should be composed of a set of non-smooth generation cost functions. Therefore, it is more realistic to treat the cost function as a set of piecewise quadratic functions as illustrated in Fig. 1, which is defined as follows [7]:

\[
F_i(P_i) = \begin{cases} 
  a_{i,1} + b_{i,1} P_i + c_{i,1} P_i^2 & \text{if } P_{i,\text{min}} \leq P_i \leq P_{i,1} \\
  a_{i,2} + b_{i,2} P_i + c_{i,2} P_i^2 & \text{if } P_{i,1} \leq P_i \leq P_{i,2} \\
  \vdots \\
  a_{i,m} + b_{i,m} P_i + c_{i,m} P_i^2 & \text{if } P_{i,m-1} \leq P_i \leq P_{i,\text{max}} 
\end{cases}
\]

where,
- \( a_{i,j}, b_{i,j}, c_{i,j} \) : cost coefficients of generator \( i \) for the \( j \)-th power level.

![Fig. 1. Piecewise quadratic and incremental cost function of a generator.](image)

III. IMPLEMENTATION OF PSO FOR ELD PROBLEMS

A. Overview of the PSO

Kennedy and Eberhart [2] developed a particle swarm optimization (PSO) algorithm based on the behavior of individuals (i.e., particles or agents) of a swarm. Its roots are in zoologist's modeling of the movement of individuals (i.e., fishes, birds, insects) within a group. It has been noticed that members of the group seem to share information among them, a fact that leads to increased efficiency of the group [13]. The PSO algorithm searches in parallel using a group of individuals similar to other AI-based heuristic optimization techniques. Each individual corresponds to a candidate solution to the problem. Individuals in a swarm approach to the optimum through its present velocity, previous experience, and the experience of its neighbors.

In a physical \( n \)-dimensional search space, the position and velocity of individual \( i \) are represented as the vectors \( X_i = (x_{i,1}, \ldots, x_{i,n}) \) and \( V_i = (v_{i,1}, \ldots, v_{i,n}) \) in the PSO algorithm. Let \( P_{\text{best},i} = (x_{i,1}^{\text{best}}, \ldots, x_{i,n}^{\text{best}}) \) and \( G_{\text{best},i} = (x_{i,1}^{\text{best}}, \ldots, x_{i,n}^{\text{best}}) \) be the best position of individual \( i \) and its neighbors' best position so far, respectively. Using the information, the updated velocity of individual \( i \) is modified under the following equation in the PSO algorithm:

\[
V_{i,t+1} = \omega V_{i,t} + c_1 \text{rand} \times (P_{\text{best},i} - X_{i,t}) \\
+ c_2 \text{rand} \times (G_{\text{best},i} - X_{i,t})
\]  

939
The dynamic process of the PSO algorithm can be summarized as follows:

1. **Initialization**
   - A group of individuals is created at random.
   - The position and velocity of each individual are initialized.

2. **Velocity and Position Update**
   - The velocity of each individual is updated using the following equation:
     \[ \mathbf{v}_i^{k+1} = w \cdot \mathbf{v}_i^k + c_1 \cdot r_1 \cdot (\mathbf{p}_{best_i} - \mathbf{x}_i^k) + c_2 \cdot r_2 \cdot (\mathbf{g}_{best} - \mathbf{x}_i^k) \]
     where:
     - \( \mathbf{v}_i^k \): velocity of individual \( i \) at iteration \( k \).
     - \( w \): weight parameter.
     - \( c_1, c_2 \): weight factors.
     - \( r_1, r_2 \): random numbers between 0 and 1.
     - \( \mathbf{x}_i^k \): position of individual \( i \) at iteration \( k \).
     - \( \mathbf{p}_{best_i} \): best position of individual \( i \) until iteration \( k \).
     - \( \mathbf{g}_{best} \): best position of the group until iteration \( k \).

3. **Update of \( \mathbf{v}_{i}^{k+1} \)**
   - If the updated velocity \( \mathbf{v}_{i}^{k+1} \) is outside the boundary, it is adjusted.

4. **Update of \( \mathbf{x}_{i}^{k+1} \)**
   - A new position is calculated by adding the modified velocity to the current position.
   - The position is then checked for constraint satisfaction.

5. **Modification of \( \mathbf{v}_i^{k+1} \)**
   - The velocity is modified to ensure it satisfies the constraints.

6. **Update of \( \mathbf{p}_{best} \)**
   - The personal best position is updated if the current position is better.

7. **Update of \( \mathbf{g}_{best} \)**
   - The group best position is updated if the current position is better.

8. **Stop the Process**
   - The process repeats until a stopping criterion is met.

After creating the initial position of each individual, the velocity of each individual is also created at random. The following strategy is used in creating the initial velocity:

\[ (p_{j,\min} - \epsilon) - p_{j}^0 \leq v_{j} \leq (p_{j,\max} + \epsilon) - p_{j}^0 \]  \hspace{1cm} (7)

where \( \epsilon \) is a small positive real number. The velocity element \( j \) of individual \( i \) is generated at random within the boundary. The developed initialization scheme always guarantees to produce individuals satisfying the constraints as well as not to deviate from the concept of the PSO algorithm. The initial \( \mathbf{p}_{best_i} \) of individual \( i \) is set as the initial position of individual \( i \) and the initial \( \mathbf{g}_{best} \) is determined as the position of an individual with minimum payoff of (1).
The weighting function is defined as follows [1], [3]:

$$\omega = \omega_{\text{max}} - \frac{\omega_{\text{max}} - \omega_{\text{min}}}{\text{Iter}_{\text{max}}} \times \text{Iter}$$

(8)

where,

- $\omega_{\text{max}}$, $\omega_{\text{min}}$: initial, final weights,
- $\text{Iter}_{\text{max}}$: maximum iteration number,
- $\text{Iter}$: current iteration number.

Also, the parameters in equations (6) and (8) are selected as follows [1], [3]:

$$c_1 = c_2 = 2.0, \quad \omega_{\text{max}} = 0.9, \quad \omega_{\text{min}} = 0.4$$

3) Position Modification Considering Constraints: The position of each individual is modified by (6) based on its updated velocity. The resulting position of an individual is not always guaranteed to satisfy the inequality constraints due to over/under velocity. If any element of an individual violates its inequality constraint due to over/under speed then the position of the individual is fixed to its maximum/minimum operating point. Fig. 3 illustrates how the position of element $j$ of individual $i$ is adjusted to its maximum when over-velocity situation occurs. The similar strategy is used for individual's position adjustment to its minimum point.

![Fig. 3. Adjustment strategy for an individual's position within boundary.](image)

Although the aforementioned method always produces the position of each individual satisfying the inequality constraints (3) and (4), the problem of equality constraint still remains to be resolved. Therefore, it is necessary to develop a new strategy such that the summation of all elements in an individual (i.e., $\sum p_j$) is equal to the total system demand. To resolve the equality constraint problem without intervening the dynamic process inherent in the PSO algorithm, we propose the following heuristic procedures:

**Step 1)** Set $j = 1$. Let present iteration be $k$.

**Step 2)** Select an element (i.e., generator) of individual $i$ at random and store in an index array $A(n)$.

**Step 3)** Modify the value of element $j$ (i.e., output of generator $j$) using (5), (6), and the position adjustment strategy to satisfy its inequality constraint as follows:

$$p_{yj}^{k+1} = \begin{cases} p_{yj}^k + v_{ij}^{k+1} & \text{if } p_{yj}^k + v_{ij}^{k+1} \leq p_{yj,\text{max}} \\ p_{yj,\text{min}} & \text{if } p_{yj}^k + v_{ij}^{k+1} < p_{yj,\text{min}} \\ p_{yj,\text{max}} & \text{if } p_{yj}^k + v_{ij}^{k+1} > p_{yj,\text{max}} \end{cases}$$

(9)

**Step 4)** If $j = n-1$ then go to Step 5, otherwise $j = j+1$ and go to Step 2.

**Step 5)** The value of the last element of individual $i$ is determined by subtracting $\sum p_j$ from $D$. If the value is not within its boundary then adjust the value using (9) and go to Step 6, otherwise go to Step 8.

**Step 6)** Set $l = 1$.

**Step 7)** Readjust the value of element $l$ in the index array $A(n)$ to the value satisfying equality condition (i.e., $D - \sum p_j$). If the value is within its boundary then go to Step 8; otherwise, change the value of element $l$ using (9). Set $l = l+1$, and go to Step 7. If $l = n+1$, go to Step 6.

**Step 8)** Stop the modification procedure.

4) Update of $P_{\text{best}}$ and $G_{\text{best}}$: The $P_{\text{best}}$ of each individual at iteration $k+1$ is updated as follows:

$$P_{\text{best}}^{k+1}_i = X_i^{k+1} \quad \text{if } TC_i^{k+1} < TC_i^k$$

$$P_{\text{best}}^{k+1}_i = P_{\text{best}}^k \quad \text{if } TC_i^{k+1} > TC_i^k$$

where,

$$TC_i : \text{the object function evaluated at the position of individual } i$$

Also, $G_{\text{best}}$ at iteration $k+1$ is set as the best evaluated position among $P_{\text{best}}^{k+1}$.

5) Stopping criteria: The proposed PSO is terminated if the iteration approaches to the predefined maximum iteration.

**IV. CASE STUDIES**

To assess the efficiency and effectiveness of the proposed PSO, it has been applied to ELD problems where the cost functions used are the quadratic and piecewise quadratic cost functions. The results obtained for the test systems are compared with those of the numerical lambda-iteration method [8], the hierarchical numerical method (HM) [7], the improved evolutionary programming (IEP) [9], and the modified Hopfield neural network (MHNN) [10].

The proposed PSO is applied to the ELD problem with 3 generators where the cost functions used are the quadratic cost functions. Table 1 shows the cost functions and the
related minimum/maximum operating points of 3 generators. Here, the system demand is 850 MW. Table II shows the comparison of the results from PSO, NM (the lambda-iteration method), IEP, and MHNN.

TABLE I

<table>
<thead>
<tr>
<th>Unit</th>
<th>$a_i$</th>
<th>$b_i$</th>
<th>$c_i$</th>
<th>$P_{CON}$</th>
<th>$P_{PUB}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>561.0</td>
<td>7.92</td>
<td>0.001562</td>
<td>150.0</td>
<td>600.0</td>
</tr>
<tr>
<td>2</td>
<td>310.0</td>
<td>7.85</td>
<td>0.00194</td>
<td>100.0</td>
<td>400.0</td>
</tr>
<tr>
<td>3</td>
<td>78.0</td>
<td>7.97</td>
<td>0.00482</td>
<td>50.0</td>
<td>200.0</td>
</tr>
</tbody>
</table>

TABLE II

<table>
<thead>
<tr>
<th>Unit</th>
<th>NM</th>
<th>MHNN</th>
<th>IEP (pop=10)</th>
<th>PSO (par=10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>393.170</td>
<td>393.800</td>
<td>393.17009</td>
<td>393.16983</td>
</tr>
<tr>
<td>2</td>
<td>334.604</td>
<td>334.100</td>
<td>334.60337</td>
<td>334.60075</td>
</tr>
<tr>
<td>3</td>
<td>122.226</td>
<td>122.300</td>
<td>122.22564</td>
<td>122.22242</td>
</tr>
<tr>
<td>TP</td>
<td>850.000</td>
<td>849.200</td>
<td>850.00000</td>
<td>850.00000</td>
</tr>
<tr>
<td>TC</td>
<td>8194.35612</td>
<td>8187.00000</td>
<td>8194.35614</td>
<td>8194.35612</td>
</tr>
</tbody>
</table>

*pop: population size, par: number of particle, TP: total power [MW], TC: total generation cost [$].

Fig. 4. Convergence characteristics of PSO for different number of particles.

As seen in the tables, the proposed PSO has always provided better solutions than IEP [9]. Also, the PSO resulted in lower generation cost compared with the hierarchical method (HM) [7]. When compared with MHNN [11], the PSO has provided better solution for the demand of 2700 MW.

The proposed PSO has been applied to the ELD problem with piecewise quadratic functions and 10 generators. The piecewise cost coefficients and related constraints of generators are given in [7,9,11]. In this case, we have changed the total system demand from 2400 MW to 2700 MW. The results from the PSO are compared with those of the hierarchical method (HM) [7], IEP [9], and MHNN [11] in Tables III and IV. Unlike to the case of the smooth cost functions, it is impossible to find the global solution with the numerical approach for the ELD problems with non-smooth cost functions.

Table III

<table>
<thead>
<tr>
<th>S</th>
<th>U</th>
<th>HM</th>
<th>MHNN</th>
<th>IEP (pop=30)</th>
<th>PSO (par=30)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>F</td>
<td>GEN</td>
<td>F</td>
<td>GEN</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>193.2</td>
<td>1</td>
<td>189.1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>204.1</td>
<td>1</td>
<td>202.0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>259.1</td>
<td>1</td>
<td>254.0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>283.4</td>
<td>3</td>
<td>239.9</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>249.0</td>
<td>1</td>
<td>241.7</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>195.5</td>
<td>1</td>
<td>233.0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>261.0</td>
<td>1</td>
<td>254.1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>234.3</td>
<td>3</td>
<td>232.9</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>325.3</td>
<td>1</td>
<td>320.0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>246.3</td>
<td>1</td>
<td>240.3</td>
<td>1</td>
</tr>
<tr>
<td>TP</td>
<td></td>
<td>2401.2</td>
<td></td>
<td>2400.0</td>
<td></td>
</tr>
<tr>
<td>TC</td>
<td></td>
<td>488.500</td>
<td></td>
<td>481.700</td>
<td></td>
</tr>
</tbody>
</table>

Table IV

<table>
<thead>
<tr>
<th>S</th>
<th>U</th>
<th>HM</th>
<th>MHNN</th>
<th>IEP (pop=30)</th>
<th>PSO (par=30)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>F</td>
<td>GEN</td>
<td>F</td>
<td>GEN</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>218.4</td>
<td>2</td>
<td>225.7</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>211.8</td>
<td>1</td>
<td>215.2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>281.0</td>
<td>1</td>
<td>291.8</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>239.7</td>
<td>3</td>
<td>242.3</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>279.0</td>
<td>1</td>
<td>293.7</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>239.7</td>
<td>3</td>
<td>242.3</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>239.0</td>
<td>3</td>
<td>232.8</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>239.7</td>
<td>3</td>
<td>242.3</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>429.2</td>
<td>1</td>
<td>355.1</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>275.2</td>
<td>1</td>
<td>288.8</td>
<td>1</td>
</tr>
<tr>
<td>TP</td>
<td></td>
<td>2702.2</td>
<td></td>
<td>2700.0</td>
<td></td>
</tr>
<tr>
<td>TC</td>
<td></td>
<td>625.180</td>
<td></td>
<td>626.240</td>
<td></td>
</tr>
</tbody>
</table>
V. CONCLUSION

This paper presents a new approach to ELD problems based on the PSO algorithm. A position adjustment strategy is incorporated in the PSO framework in order to provide solutions satisfying the inequality constraints. The equality constraint in the ELD problem is resolved by reducing the degree of freedom by one at random. The strategies for handling constraints are devised in order not to intervene the dynamic process of PSO algorithm.

The proposed PSO provided the global solution for the ELD problems with smooth cost functions within a reasonable computation time and iteration number. Also, it was better compared to the existing heuristic optimization techniques in terms of generation cost or constraints satisfaction.

In the case of non-smooth function problems, the proposed PSO has shown superiority to the conventional numerical method and the evolutionary programming approach. Also we have observed that it has provided very similar results with the modified Hopfield neural network.

VI. ACKNOWLEDGMENT

This work is supported by the Korea Ministry of Commerce, Industry and Energy (MOCIE) and Korea Electric Power Research Institute (KEPRI) through MIERC program.

VII. REFERENCES