Short-Term Hydrothermal Scheduling using Imperialist Competitive Algorithm


Department of Electrical Engineering, Benha University Egypt.

*Corresponding author: mohaayoub2018@gmail.com

Abstract: In this study, the Imperialist Competitive Algorithm (ICA) is proposed to solve a multi-chain Short-Term Hydrothermal Scheduling problem (STHTS). It aims to minimize the generation cost of the thermal plants while satisfying the thermal and hydro plants constraints. In order to evaluate the effectiveness of the ICA, it has been tested on a system having a hydro plant with four-cascaded reservoir and a thermal plant. The results are compared with that obtained by other techniques. The ICA has good convergence and the better results.

Keywords: Imperialist Competitive Algorithm, Short-Term Hydrothermal Scheduling.

Article History: received 6 January 2021; accepted 20 April 2021; published 30 April 2021.

INTRODUCTION

Short-Term Hydrothermal Scheduling (STHTS) is the manner in which the fuel cost of the thermal plant is minimized for the economical operation of hydrothermal systems. It aims to get optimal operation scheduling involving the hydro and thermal constraints with minimizing the operational cost of the thermal units. The constraints of hydrothermal system include generation limits, water balance, power balance, the physical restrictions on the reservoir storage and discharge capacity. Therefore, STHTS is a non-smooth, non-linear, non-convex and large-scale optimization problem.

In recent years, Evolutionary Algorithms (EA) have been widely used due to their flexibility, versatility, and robustness in searching a globally optimal solution. Several Evolutionary Techniques, such as Genetic Algorithm (GA)[1], Evolutionary Programming (EP), Couple Based Particle Swarm Optimization (CPSO)[2], Honey Bee Mating Optimization (HBMO)[3], Differential Evolution (DE)[4], Modified Differential Evolution (MDE)[5], Symbiotic Organisms Search (SOS)[6, 7], Teaching Learning Based Optimization (TLBO)[8], Crisscross Optimization (CSO)[9], and Modified Dynamic Neighborhood Learning Based Particle Swarm Optimization (MDNLPSO)[10] have been intended for scheduling of the hydrothermal problem.

This paper recommended new algorithm known as Imperialist Competitive Algorithm (ICA). The basis of ICA began from the attempting of the world countries to increase their power over the different countries for employing their resources and support their own government. Imperialist countries tried to state their power over the different countries to be from their colonies. Furthermore, compete with each other to pick the ownership of the other countries. During this process, empires that have more powerful will gain more power and inefficient ones will finally slump. ICA attempt to model this procedure to obtain the best solution. Up to now, the ICA informed a good result in finding a global solution that encourage to use this algorithm in optimization problems in many applications [11-14].

PROBLEM FORMULATION

The main objective of Short-Term Hydrothermal Scheduling (STHTS) problem is to expose the optimal schedule of thermal and hydro plants to reduce the total production as possible while fulfilling all constraints. The total production cost function is usually expressed as

\[
\min TC = \sum_{j=1}^{N_g} \sum_{t=1}^{T} \left( a_{j} p_{gj}^t + b_{j} p_{gj}^t + c_{j} \right) 
\]

(1)

Where TC is the total production cost, T is the number of scheduling time intervals, \( N_g \) is the thermal generation plants, \( P_{gj} \) is the thermal output power of the \( j \)th unit at \( t \)th time interval and \( a_j, b_j \), and \( c_j \) are the \( j \)th thermal plant cost coefficients.

2.1 System load balance

The system load balance equation can be given with neglecting the transmission losses as:

\[
\sum_{i=1}^{N_h} \sum_{t=1}^{T} p_{h_i}^t = \sum_{j=1}^{N_g} \sum_{t=1}^{T} p_{gj}^t
\]

(2)

Where \( N_h \) is the number of hydro plants, \( P_{h_i}^t \) is output of the \( i \)th hydro unit at \( t \)th time interval, \( P_{gj}^t \) is the total load demand at the \( t \)th time interval.

2.2 Thermal plant generation limits

\[
P_{gj}^{\min} \leq P_{gj}^t \leq P_{gj}^{\max}
\]

(3)

Where, \( P_{gj}^{\min} \), \( P_{gj}^{\max} \) are minimum and maximum limits of the \( j \)th thermal plant.
2.3 Hydro plant generation limits

\[ P_{hi}^{\text{min}} \leq P_{hi} \leq P_{hi}^{\text{max}} \]  

(4)

Where, \( P_{hi}^{\text{min}} \) and \( P_{hi}^{\text{max}} \) are minimum and maximum limits of the \( i^{th} \) hydro plant.

2.4 Hydro plant power generation

The hydro plant power generation is a function of water discharge rate through turbine and reservoir storage volume and it expressed as:

\[ P_{hi} = C_{1i} V_{ji}^{2} + C_{2i} q_{ji} + C_{3i} V_{j} + C_{4i} V_{j} + C_{5i} V_{j} + C_{6i} \]  

(5)

Where, \( C_{1i}:C_{6i} \) are the generation coefficients of the \( i^{th} \) hydro plant. \( V_{ji} \) is the reservoir storage volume level of \( j^{th} \) reservoir at \( t^{th} \) time interval. \( q_{ji} \) is water discharge rate of \( j^{th} \) reservoir at \( t^{th} \) time interval.

2.5 Hydro plant water discharge limits

\[ q_{ji}^{\text{min}} \leq q_{ji} \leq q_{ji}^{\text{max}} \]  

(6)

Where, \( q_{ji}^{\text{min}} \) and \( q_{ji}^{\text{max}} \) are the minimum and maximum water discharge limits of \( i^{th} \) hydro plant.

2.6 Reservoir storage volume limits

\[ V_{ji}^{\text{min}} \leq V_{ji} \leq V_{ji}^{\text{max}} \]  

(7)

Where, \( V_{ji}^{\text{min}} \) and \( V_{ji}^{\text{max}} \) are the minimum and maximum limits of reservoir storage volume of \( i^{th} \) hydro plant.

2.7 Water dynamic balance equation

\[ V_{ji}^{t} = V_{ji}^{t-1} + I_{ji}^{t} - q_{ji}^{t} - \sum_{m=1}^{N_{ui}} q_{jm}^{t-r_{jm}^{m}} \]  

(8)

Where \( I_{ji}^{t} \) is the water inflow rate of \( j^{th} \) reservoir at \( t^{th} \) time interval, \( r_{jm}^{m} \) is the time delay between the hydro plant and its upstream plants at \( t^{th} \) interval, \( N_{u} \) number of upstream plants above to the \( i^{th} \) hydro plant.

2.8 Initial and final reservoir storage volume limits

\[ V_{ji}^{\text{init}} = V_{ji}^{\text{init} \_i} ; V_{ji}^{\text{end}} = V_{ji}^{\text{end} \_i} \]  

(9)

Where, \( V_{ji}^{\text{init}} \) and \( V_{ji}^{\text{end}} \) are the initial and final reservoir storage volume of \( j^{th} \) reservoir.

\( V_{ji}^{\text{in}}, V_{ji}^{\text{end}} \) are the \( j^{th} \) reservoir storage volume at the beginning and ending of the time horizon, respectively.

AN OVERVIEW OF IMPERIALIST COMPETITIVE ALGORITHM AND IMPLEMENTATION TO THE PROBLEM

The rule of ICA was made by Atashpaz-Gargari and Lucas (2007) [14]. The ICA simulated the social political development of imperialism and compete the imperialist. ICA began with initial population called countries that were divided into two kinds of countries; the strongest countries were chosen to be the imperialist and the residual countries establish the colonies of these imperialists. The colonies of initial countries were distributed among the imperialists constructed on the imperialist's power. All the imperialists and their colonies united together to form empires.

After empires formation, the competition between empires begin. The weaker empires miss out their colonies to the strong empires until we reach one empire and the other rest of their colonies. This empire represents the final best solution. The application steps of ICA algorithm on (STHTS) problem are shown on Figure 1. And the algorithm is divided into the following five stages:

3.1 Initializing phase:

Firstly, Preparation of initial populations. Each solution (i.e., country) in form of an array as (10).

\[ X = [P_{1}, P_{2}, P_{3}, \ldots, P_{N}]. \]  

(10)

Where \( P \) represent a variables, and \( N_{v} \) n–dimension of the optimized problem. The cost function the countries can depicted as (11).

\[ \text{Cost} = f(\text{country}) = f(\text{var}) \]  

(11)

Then Initializing the empires with initial populations \( (N_{p}) \) involved two types of countries [i.e., colony \( (N_{c}) \) and imperialist \( (N_{i}) \)] which together form the empires. The normalized cost \( C_{n} \) of an imperialist was depicted as.

\[ C_{n} = c_{n} - \max \{c_{i}\} \]  

(12)

Whereas, \( c_{n} \) was the \( n^{th} \) imperialist cost.

The colonies of initial countries were distributed among the imperialists constructed on the imperialist's power. The normalized power \( P_{n} \) of each imperialist was depicted as.

\[ P_{n} = \left| \frac{C_{n}}{\sum_{i=1}^{N_{c}} C_{i}} \right| \]  

(13)

3.2 Moving phase:

Colonies moved toward their imperialist with \( x \) units as depicted in (14).

\[ x : U(0, \beta \times d) \]  

(14)

Where \( x \) is a random variable with uniform distribution, \( \beta \) is a number greater than 1, and \( d \) is the distance between an imperialist and its colony. And the direction of movement of colonies were depicted as (15).

\[ \theta : U(-\gamma, \gamma) \]  

(15)

Where \( \theta \) was a random variable with uniform distribution, and \( \gamma \) was a parameter that regulated the change from the original direction.
3.3 Exchanging phase:
During movement of colonies, if the new situation of the colony is better (based on the cost function) than the corresponding imperialist so imperialist and the colony change their positions.

3.4 Competition phase:
Firstly, calculated the total power of an empire as (16).

\[ T.C_n = \text{cost (imperialist}_n) + \xi \text{mean (cost (colonies of empire}_n)) \] (16)

Where, \( T.C_n \) was the total power of the \( n \)th empire and \( \xi \) is a positive number less than 1.

Then all empires were tried to acquire more colonies from other empires. The weakest empires loose it's colonies during competition between them.

3.5 Eliminating phase:
The empires that lose their colonies were collapsed and eliminated. Finally, all the colonies will be under the dominance of the most powerful empire.

CASE STUDY AND NUMERICAL RESULTS
To examine the flexibility of the ICA to find the optimal solution of STHTS problem, the ICA algorithm has been applied on a case study and also that the results were compared with different algorithms within the literature. The program was processed using MATLAB 2016 on a Pentium i3 laptop computer, 2.53 GHz processor speed and 4 GB RAM.

4.1 Test System
This test system considers a multi-chain cascade of four hydro plants and a thermal plant represented by an equivalent thermal plant including power balance, water balance, generation limits, the physical restrictions on the discharge capacity and reservoir storage. The hydro system is shown in Figure 2. In this case, the fuel cost of the equivalent thermal plant is considered to be of quadratic in nature, i.e., the valve point effect has been neglected. The total scheduling period is 1 day which has been divided into 24 intervals all data are taken from [15].

Table 1 Show the optimal hydro discharges for this system, and Table 2 Shows the individual hydro power and thermal power generated and Table 3 shows the comparison of best and worst cost as well as the computation time among ICA and other methods. It show that some of other algorithms have less time solution than ICA, but still its simulation time is good enough. The cascaded reservoir storage volumes for the test system is presented in Figure 3. The convergence curve of the proposed algorithm is given in Figure 4 and show that ICA algorithm reach to optimal solution with very small number of iterations.
Table 2. Hydro and thermal generation power

<table>
<thead>
<tr>
<th></th>
<th>Hydro Power Gen. (MW)</th>
<th>Thermal Power Gen. (MW)</th>
<th>Power Demand (MW)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$P_{h1}$</td>
<td>$P_{h2}$</td>
<td>$P_{h3}$</td>
</tr>
<tr>
<td>1</td>
<td>97.65151</td>
<td>81.27684</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>93.4365</td>
<td>68.61934</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>71.47132</td>
<td>74.25258</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>88.88353</td>
<td>80.72279</td>
<td>35.3986</td>
</tr>
<tr>
<td>5</td>
<td>76.25723</td>
<td>51.63139</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>84.11946</td>
<td>56.75525</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>81.55369</td>
<td>69.10467</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>54.85934</td>
<td>54.37668</td>
<td>51.27365</td>
</tr>
<tr>
<td>9</td>
<td>80.08311</td>
<td>67.32385</td>
<td>28.51075</td>
</tr>
<tr>
<td>10</td>
<td>85.9503</td>
<td>65.44247</td>
<td>37.84124</td>
</tr>
<tr>
<td>11</td>
<td>60.50938</td>
<td>68.3543</td>
<td>48.73976</td>
</tr>
<tr>
<td>12</td>
<td>61.02126</td>
<td>67.14944</td>
<td>48.75015</td>
</tr>
<tr>
<td>13</td>
<td>52.49306</td>
<td>45.63351</td>
<td>54.04749</td>
</tr>
<tr>
<td>14</td>
<td>79.63589</td>
<td>38.15865</td>
<td>56.64283</td>
</tr>
<tr>
<td>15</td>
<td>84.33989</td>
<td>58.39447</td>
<td>43.7102</td>
</tr>
<tr>
<td>16</td>
<td>77.96906</td>
<td>59.37094</td>
<td>41.12587</td>
</tr>
<tr>
<td>17</td>
<td>88.19862</td>
<td>36.87301</td>
<td>58.74001</td>
</tr>
<tr>
<td>18</td>
<td>59.47086</td>
<td>36.87301</td>
<td>59.83014</td>
</tr>
<tr>
<td>19</td>
<td>54.20861</td>
<td>37.60531</td>
<td>61.19554</td>
</tr>
<tr>
<td>20</td>
<td>54.32849</td>
<td>39.0459</td>
<td>28.19569</td>
</tr>
<tr>
<td>21</td>
<td>62.18572</td>
<td>43.31517</td>
<td>52.42664</td>
</tr>
<tr>
<td>22</td>
<td>74.22821</td>
<td>42.93958</td>
<td>56.18937</td>
</tr>
<tr>
<td>23</td>
<td>67.75848</td>
<td>53.98338</td>
<td>51.91712</td>
</tr>
<tr>
<td>24</td>
<td>59.87987</td>
<td>59.63356</td>
<td>56.04564</td>
</tr>
</tbody>
</table>

Table 3. Comparison of performance

<table>
<thead>
<tr>
<th>Methods</th>
<th>Min cost</th>
<th>Max cost</th>
<th>Average cost</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDNLPSO [10]</td>
<td>922336.3</td>
<td>923404.5</td>
<td>922676.2</td>
<td>35</td>
</tr>
<tr>
<td>DNLPSO [10]</td>
<td>922,498</td>
<td>923,580</td>
<td>922,837</td>
<td>37</td>
</tr>
<tr>
<td>SOS [6]</td>
<td>922295.25124</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CSO [9]</td>
<td>922316.16</td>
<td>922373.10</td>
<td>922326.70</td>
<td>36</td>
</tr>
<tr>
<td>CPSO [2]</td>
<td>922,328.64</td>
<td>922,508.67</td>
<td>922,346.33</td>
<td>20.8</td>
</tr>
<tr>
<td>HBMO [3]</td>
<td>923,300</td>
<td>927790</td>
<td>925905</td>
<td>153.8</td>
</tr>
<tr>
<td>ICA</td>
<td>921612.3478</td>
<td>926810.09</td>
<td>924211.17</td>
<td>53.19</td>
</tr>
</tbody>
</table>

Figure 3. Reservoir storage volumes for Test system with ICA
5. CONCLUSIONS

Short Term Hydrothermal Scheduling problem (STHTS) has been solved by imperialist competitive algorithm. (STHTS) Considered one of non-linear and non-convex optimization problems in power systems. The proposed ICA algorithm has been successfully applied on a test system that consists four hydro and thermal plant. While satisfying the operational constraints, minimization of the fuel cost that is aimed in this problem. The results have been compared with similar algorithms. Reported in the literature, it is observed from the test results and comparisons that the proposed ICA algorithm performs well in solving hydrothermal scheduling problems. The minimum cost by ICA, MDNLPSO and SOS are 921612.3478, 922336.3, and 922295.2512, respectively. It is seen that ICA present a good result and can be used for solving other complex engineering problems.

REFERENCES


