Augmented Lagrangian Hopfield Network for Combined Heat and Power Economic Dispatch

Vo Ngoc Dieu and Weerakorn Ongsakul

Abstract—This paper proposes an augmented Lagrangian Hopfield network (ALHN) for combined heat and power economic dispatch (CHPED) problem. The ALHN is the continuous Hopfield neural network based on augmented Lagrangian relaxation as its energy function. In the proposed ALHN, its energy function is augmented by Hopfield terms from Hopfield neural network and penalty factors from augmented Lagrangian relaxation which help to damp out oscillation of the neural network during transient process. Consequently, the proposed ALHN obtains fast convergence to optimal solution for the complicated CHPED problem. The proposed ALHN has been tested on various systems and results are compared to Lagrangian relaxation (LR), genetic algorithm (GA), improved ant colony search algorithm (IACS), evolutionary programming (EP), and improved genetic algorithm with multiplier updating (IGA-MU). The results show that the proposed neural network has more advantages than other methods in terms of total costs and computational times, especially for large-scale systems of the CHPED problem.

Keywords—Augmented Lagrangian Hopfield network, combined heat and power, economic dispatch.

NOMENCLATURE

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Np</td>
<td>number of pure power generating units;</td>
</tr>
<tr>
<td>Nc</td>
<td>number of co-generation units;</td>
</tr>
<tr>
<td>Nw</td>
<td>number of pure heat production units;</td>
</tr>
<tr>
<td>Nmax</td>
<td>maximum number of iterations of neural network;</td>
</tr>
<tr>
<td>i</td>
<td>index of pure power generating units;</td>
</tr>
<tr>
<td>j</td>
<td>index of co-generation units;</td>
</tr>
<tr>
<td>k</td>
<td>index of pure heat production units;</td>
</tr>
<tr>
<td>F_i(P_i)</td>
<td>cost function of pure power generating unit i;</td>
</tr>
<tr>
<td>F_j(H_j)</td>
<td>cost function of pure heat production units k;</td>
</tr>
<tr>
<td>P_i</td>
<td>output power of pure power generating unit i, in MW;</td>
</tr>
<tr>
<td>P_j</td>
<td>output power of co-generation unit j, in MW;</td>
</tr>
<tr>
<td>H_j</td>
<td>heat production of co-generation unit j, in MWh;</td>
</tr>
<tr>
<td>H_k</td>
<td>heat production of pure heat production unit k, in MWh;</td>
</tr>
<tr>
<td>P_D</td>
<td>system power demand, in MW;</td>
</tr>
<tr>
<td>H_D</td>
<td>system heat demand, in MWh;</td>
</tr>
<tr>
<td>P_i^max</td>
<td>maximum output of pure power unit i, in MW;</td>
</tr>
<tr>
<td>P_i^min</td>
<td>minimum output of pure power unit i, in MW;</td>
</tr>
<tr>
<td>P_j^max</td>
<td>maximum output power of co-generation unit j, a function of heat production, in MW;</td>
</tr>
<tr>
<td>P_j^min</td>
<td>minimum output power of co-generation unit j, a function of heat production, in MW;</td>
</tr>
<tr>
<td>H_j^max</td>
<td>maximum heat output of co-generation unit j, a function of output power, in MWh;</td>
</tr>
<tr>
<td>H_j^min</td>
<td>minimum heat output of co-generation unit j, a function of output power, in MWh;</td>
</tr>
<tr>
<td>H_k^max</td>
<td>maximum output of pure heat production unit k, in MWh;</td>
</tr>
<tr>
<td>H_k^min</td>
<td>minimum output of pure heat production unit k, in MWh;</td>
</tr>
<tr>
<td>λp, λh</td>
<td>Lagrange multipliers for power and heat demand balances, respectively;</td>
</tr>
<tr>
<td>βp, βh</td>
<td>penalty factors for power and heat demand balances, respectively;</td>
</tr>
<tr>
<td>V_{pi}</td>
<td>output of continuous neuron for pure power generating unit i representing P_i;</td>
</tr>
<tr>
<td>V_{pj}</td>
<td>output of continuous neuron for co-generation unit j representing P_j;</td>
</tr>
<tr>
<td>V_{hj}</td>
<td>output of continuous neuron for co-generation unit j representing H_j;</td>
</tr>
<tr>
<td>V_{hk}</td>
<td>output of continuous neuron for pure heat production unit k representing H_k;</td>
</tr>
<tr>
<td>V_{lp}</td>
<td>output of multiplier neuron for power demand balance representing λ_p;</td>
</tr>
<tr>
<td>V_{lh}</td>
<td>output of multiplier neuron for heat demand balance representing λ_h;</td>
</tr>
<tr>
<td>U_{pi}</td>
<td>input of continuous neuron for pure power generating unit i corresponding to output V_{pi};</td>
</tr>
<tr>
<td>U_{pj}</td>
<td>input of continuous neuron for co-generation unit j corresponding to output V_{pj};</td>
</tr>
<tr>
<td>U_{hj}</td>
<td>input of continuous neuron for pure heat production unit k corresponding to output V_{hk};</td>
</tr>
<tr>
<td>U_{hk}</td>
<td>input of continuous neuron for pure heat production unit k corresponding to output V_{hk};</td>
</tr>
<tr>
<td>U_{lp}</td>
<td>input of multiplier neuron for power demand;</td>
</tr>
<tr>
<td>U_{lp}</td>
<td>input of multiplier neuron for heat demand;</td>
</tr>
<tr>
<td>g'(V)</td>
<td>inverse sigmoid function of continuous neurons;</td>
</tr>
<tr>
<td>σ</td>
<td>slope of sigmoid function of continuous neurons;</td>
</tr>
<tr>
<td>α_{pi}, α_{pj}</td>
<td>continuous neuron updating step sizes for power generation;</td>
</tr>
<tr>
<td>α_{pi}, α_{pj}</td>
<td>continuous neuron updating step sizes for heat production;</td>
</tr>
</tbody>
</table>
| α_{lp}, α_{lh} | multiplier neuron updating step sizes.

1. INTRODUCTION

Combined heat and power generation (co-generation) units have an increasingly important role in energy production technology recently [1]-[3]. A co-generation unit can provide not only power but also heat to customers. For most co-generation units, there is a mutual dependency between heat and power, e.g. the heat production capacity depends on power generation and
vice versa. Therefore, the combined heat and power economic dispatch (CHPED) problem introduces complexities in the integration of co-generation units into the power system economic dispatch since both the power demand and the heat demand must be satisfied.

To date, although the combined heat and power systems are popular, only few works have been reported in the literature in the area of CHPED problems. An exploitation of the high degree of separability of the cost function and the constraint is applied in [4]. The method has proved to converge faster than conventional procedure based on quadratic programming. A method based on two dimensional probability load density functions for performing probabilistic production simulation involving combined heat and power units is developed in [5]. In this method, the equivalent load functions, the expected energy generation of units, the expected unserved energy, and the expected overflow are determined by convolution of the combined heat and power units. In [6], the CHPED problem is decomposed into two sub-problems, the heat dispatch and the power dispatch, connected through the heat-power feasible region constraints of co-generation units. The proposed method is developed by a two-layer algorithm with the outer layer for power dispatch and the inner layer for heat dispatch. The unit heat capacities for inner layer solution by gradient searching method is passed from the outer layer by Lagrangian relaxation technique, and in mutual dependency the binding constraints of the heat are fed back to the outer layer to move the problem toward a global optimal solution. An improved penalty function for genetic algorithm (GA) to solve the CHPED problem is employed in [7]. An improved ant colony search algorithm (IACS) with positive feedback, distributed computation and the use of constructive greedy heuristic incorporated with other search techniques is proposed for the CHPED problem in [8]. The method has advantages of rapid discovery of goods solutions, avoidance of premature convergence and acceptable solutions in the early stages of the search process. However, solution obtained by this method is near global optimum. An evolutionary programming (EP) for the CHPED is developed in [9]. In this method, methods for ensuring the satisfaction of the power and heat demands and a method for determining the dispatch order of the units in the cogeneration system are developed and included in the algorithm. An improved genetic algorithm with multiplier updating (IGA-MU) for the CHPED problem is proposed in [10]. In this method, the improved genetic algorithm equipped with an improved evolutionary direction operator and migration operation is incorporated with multiplier updating method which is introduced to avoid deforming of augmented Lagrange function. The hybrid system achieves the merit order of automatically adjusting the randomly given penalty to a power value and requiring only a small size population for the problem. However, the number of iterations is still high and consequently convergence is slow.

This paper proposes an augmented Lagrange Hopfield network (ALHN) for large scale CHPED problem. The ALHN consists of continuous Hopfield neural network with energy function based on augmented Lagrangian relaxation function. The proposed ALHN is augmented by Hopfield terms from Hopfield network and penalty factor from Lagrangian relaxation which helps to damp out oscillation of Hopfield network during transient process. Thus, the convergence of the network is improved. The effectiveness of the proposed ALHN is demonstrated by comparing to LR [6], GA [7], IACS [8], EP [9], and IGA-MU [10].

The rest of the paper is organized as follows. Section 2 describes the CHPED problem formulation. The ALHN approach to CHPED problem is addressed in Section 3. Numerical results are shown in Section 4. Finally, conclusion is given in Section 5.

2. CHPED PROBLEM FORMULATION

The objective of the CHPED problem is to minimize the total operation cost of power and heat production satisfying both power and heat load demands.

The system has three types of units including pure power, combined power and heat, and pure heat units. The heat-power feasible operation region of a combined power and heat unit is shown in Fig. 1, where the boundary curve ABCDEF determines the feasible region. Along the boundary there is a trade-off between power and heat production from the unit. It can be seen that along the curve AB the unit reaches maximum output power. In contrast, the unit reaches maximum heat production along the curve CD.

Mathematically, the problem is formulated as follows

\[ \text{Min} \left\{ F(P_i) + \sum_{j=1}^{N} F_j(P_j, H_j) + \sum_{k=1}^{N} F_k(H_k) \right\} \]  

subject to

(a) power balance constraint

\[ P_D - \sum_{i=1}^{N} P_i - \sum_{j=1}^{N} P_j = 0 \]  

(b) heat balance constraint

\[ H_D - \sum_{j=1}^{N} H_j - \sum_{k=1}^{N} H_k = 0 \]  

(c) generation limit constraints

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

\[ H_{j\text{min}} \leq H_j \leq H_{j\text{max}} \]  

\[ P_{j\text{min}} \leq P_j \leq P_{j\text{max}} \]  

\[ H_{k\text{min}} \leq H_k \leq H_{k\text{max}} \]

where \( F(P_i), F_j(P_j, H_j) \) and \( F_k(H_k) \) are convex functions.

Fig. 1. Heat-power feasible region for co-generation units.

For the co-generation units, power generation limits are functions of the unit heat production and vice versa. It is obvious that the CHPED problem has more constraints than a conventional pure power economic dispatch problem. Thus, CHPED is a nonlinear, highly constrained optimization problem.

3. ALHN APPROACH TO CHPED

Initially, energy function of the ALHN is constructed based
on the augmented Lagrangian function with augmentation of Hopfield terms.

Augmented Lagrangian relaxation for the problem is formulated as follows:

\[ L = \sum_{i=1}^{N_p} F_i(P_i) + \sum_{j=1}^{N_h} F_j(P_j, H_j) + \sum_{k=1}^{N_h} F_k(H_k) \]

+ \lambda_p \left( P_D - \sum_{i=1}^{N_p} P_i - \sum_{j=1}^{N_h} P_j \right) + \frac{1}{2} \beta_p \left( P_D - \sum_{i=1}^{N_p} P_i - \sum_{j=1}^{N_h} P_j \right)^2 \tag{8} \]

+ \lambda_h \left( H_D - \sum_{j=1}^{N_h} H_j - \sum_{k=1}^{N_h} H_k \right) + \frac{1}{2} \beta_h \left( H_D - \sum_{j=1}^{N_h} H_j - \sum_{k=1}^{N_h} H_k \right)^2 \]

To implement ALHN, \( N_p \) continuous neurons for output power from the pure power units, \( N_h \) continuous neurons for output power from the co-generation units, \( N_i \) continuous neurons for heat production from the co-generation units, \( N_c \) continuous neurons for the pure heat units, one multiplier neuron for power balance and one multiplier neuron for heat demand balance are required.

Energy function of the ALHN based on the Lagrangian function is defined as follows:

\[ E = \sum_{i=1}^{N_p} F_i(V_{pi}) + \sum_{j=1}^{N_h} F_j(V_{pj}, V_{hj}) + \sum_{k=1}^{N_c} F_k(V_{ck}) \]

+ \sum_{j=1}^{N_h} V_{pj} \left( P_D - \sum_{i=1}^{N_p} V_{pi} - \sum_{j=1}^{N_h} V_{pj} \right) \]

+ \frac{1}{2} \beta_p \left( P_D - \sum_{i=1}^{N_p} V_{pi} - \sum_{j=1}^{N_h} V_{pj} \right)^2 \]

+ \sum_{j=1}^{N_h} V_{hj} \left( H_D - \sum_{j=1}^{N_h} V_{hj} - \sum_{k=1}^{N_h} V_{hj} \right) \]

+ \frac{1}{2} \beta_h \left( H_D - \sum_{j=1}^{N_h} V_{hj} - \sum_{k=1}^{N_h} V_{hj} \right)^2 \]

+ \sum_{j=1}^{N_h} \int \frac{g(V_{pj})}{2} dV + \sum_{j=1}^{N_h} \int \frac{g(V_{pj})}{2} dV \]

+ \sum_{k=1}^{N_c} \int \frac{g(V_{ck})}{2} dV + \sum_{j=1}^{N_h} \int \frac{g(V_{hj})}{2} dV \]

The four last terms in (9) are Hopfield terms where their global effect is a displacement of solutions toward the interior of the state space [12].

The dynamics of the ALHN for updating neuron inputs are derived as follows:

\[ \frac{dU_{pi}}{dt} = - \frac{\partial E}{\partial V_{pi}} = \begin{bmatrix} \partial F_i(V_{pi}) \\ \partial V_{pi} \end{bmatrix} \] + \begin{bmatrix} \lambda_p \left( P_D - \sum_{i=1}^{N_p} V_{pi} - \sum_{j=1}^{N_h} V_{pj} \right) \]

+ \begin{bmatrix} V_{pi} + \beta_p \left( P_D - \sum_{i=1}^{N_p} V_{pi} - \sum_{j=1}^{N_h} V_{pj} \right) \]

\] + \begin{bmatrix} \lambda_h \left( H_D - \sum_{j=1}^{N_h} V_{hj} - \sum_{k=1}^{N_h} V_{hj} \right) \]

+ \begin{bmatrix} V_{hj} + \beta_h \left( H_D - \sum_{j=1}^{N_h} V_{hj} - \sum_{k=1}^{N_h} V_{hj} \right) \]

\] + \begin{bmatrix} \lambda_c \left( \sum_{k=1}^{N_c} V_{ck} \right) \]

+ \begin{bmatrix} \lambda_{ck} \left( \sum_{k=1}^{N_c} V_{ck} \right) \]

Algorithm for updating inputs of neurons based on the dynamics at iteration \( \lambda \) as follows:

\[ U_{pi}^{(\lambda)} = U_{pi}^{(\lambda-1)} - \alpha_{pi} \frac{\partial E}{\partial V_{pi}} \]

\[ U_{pj}^{(\lambda)} = U_{pj}^{(\lambda-1)} - \alpha_{pj} \frac{\partial E}{\partial V_{pj}} \]

\[ U_{hj}^{(\lambda)} = U_{hj}^{(\lambda-1)} - \alpha_{hj} \frac{\partial E}{\partial V_{hj}} \]

\[ U_{ck}^{(\lambda)} = U_{ck}^{(\lambda-1)} - \alpha_{ck} \frac{\partial E}{\partial V_{ck}} \]

\[ U_{ch}^{(\lambda)} = U_{ch}^{(\lambda-1)} + \alpha_{ch} \frac{\partial E}{\partial V_{ch}} \]

The sigmoid function is defined for relationship between inputs and outputs of continuous neurons limited by maximum and minimum values as follows [13]:

\[ V_{pi} = g(U_{pi}) = \left( P_{i\text{max}} - P_{i\text{min}} \right) \frac{1 + \tanh(\sigma U_{pi})}{2} + P_{i\text{min}} \]

(22)
The energy function (9) to be minimized with respect to continuous neurons by doing the same manner.

On the other hand, the dynamics (14) cause the energy function (9) to be maximized with respect to multiplier neurons illustrated as followings

\[
\frac{dE}{dt} = \frac{\partial E}{\partial V_{ip}} \frac{dV_{ip}}{dt}
\]

Substituting (26) into (32):

\[
\frac{dE}{dt} = \frac{\partial E}{\partial V_{ip}} \frac{dU_{ip}}{dt}
\]

Substituting (14) into (34):

\[
\frac{dE}{dt} = \left( \frac{dU_{ip}}{dt} \right)^2
\]

It is obvious that the right hand side of equation (35) is always positive, so the energy function (9) is maximized when multiplier neurons status is changed. Similarly, the dynamics of multiplier neurons in (15) also cause the energy function (9) to be maximized by doing the same manner. Therefore, the energy function (9) is minimized respect to continuous neurons and maximized with respect to multiplier neurons.

Selection of parameters

The algorithm requires initial conditions for the inputs of all neurons. For the continuous neurons, the inputs are initiated by the initial value of each unit is proportion with its maximum contribution in total load demand. The initial values of continuous neurons are determined.

\[
\begin{align*}
V_{pi}^{(0)} &= \frac{P_i^{max}}{\sum_{j=1}^{N_p} P_j^{max} + \sum_{j=1}^{N_d} P_j^{max}} P_D \\
V_{pj}^{(0)} &= \frac{P_j^{max}}{\sum_{i=1}^{N_p} P_i^{max} + \sum_{i=1}^{N_d} P_i^{max}} P_D \\
V_{bj}^{(0)} &= \frac{H_j^{max}}{\sum_{i=1}^{N_j} H_i^{max} + \sum_{i=1}^{N_d} H_j^{max}} H_D \\
V_{bk}^{(0)} &= \frac{H_k^{max}}{\sum_{j=1}^{N_k} H_j^{max} + \sum_{j=1}^{N_d} H_k^{max}} H_D
\end{align*}
\]

The multiplier neurons are initialized by mean value according to simplified \( \frac{\partial E}{\partial V_{ij}} = 0 \) and \( \frac{\partial E}{\partial V_{bj}} = 0 \), in which inputs and penalty factors are neglected. The mean values multiplier neurons are obtained:

Fig 2. Sigmoid function of continuous neurons with different values of the slope.

To illustrate how the dynamic (10) cause the energy function (9) to be minimized with respect to continuous neurons, consider the effect of the change in continuous neuron \( i \) on energy function:

\[
\frac{dE}{dt} = \frac{\partial E}{\partial V_{pi}} \frac{dV_{pi}}{dt}
\]

Substituting (22) into (28):

\[
\frac{dE}{dt} = \frac{\partial E}{\partial V_{pi}} \frac{dU_{ip}}{dt}
\]

Substituting (10) into (30):

\[
\frac{dE}{dt} = \left( \frac{dU_{ip}}{dt} \right)^2
\]

Since \( g(U_{ip}) \) is a monotonic increasing function, the value of \( dE/dt \) is always negative. Therefore, the energy function (9) is minimized with respect to continuous neuron \( i \) when this neuron change its status. Similarly, the dynamics (11)-(13) also cause
\[ V_{kj}^{(0)} = \frac{\partial F_I(V_{pi}^{(0)},V_{pj}^{(0)})}{\partial V_{kj}} \]  
(40)

\[ V_{hi}^{(0)} = \frac{\partial F_I(V_{pi}^{(0)},V_{jh}^{(0)})}{\partial V_{hi}} \]  
(41)

**Termination criteria**

In the proposed ALHN, the errors are calculated from the following elements.

Power constraint error at iteration \( n \) is determined:

\[ \Delta P = P_D - \sum_{i=1}^{N_p} V_{pi}^{(n)} - \sum_{j=1}^{N_j} V_{pj}^{(n)} \]  
(42)

Heat constraint error at iteration \( n \) is determined:

\[ \Delta H = H_D - \sum_{j=1}^{N_j} V_{hj}^{(n)} - \sum_{k=1}^{N_h} V_{hk}^{(n)} \]  
(43)

Iterative errors of neurons at iteration \( n \) are defined:

\[ \Delta V_{pi} = V_{pi}^{(n)} - V_{pi}^{(n-1)} \]  
(44)

\[ \Delta V_{pj} = V_{pj}^{(n)} - V_{pj}^{(n-1)} \]  
(45)

\[ \Delta V_{hj} = V_{hj}^{(n)} - V_{hj}^{(n-1)} \]  
(46)

\[ \Delta V_{hk} = V_{hk}^{(n)} - V_{hk}^{(n-1)} \]  
(47)

\[ \Delta V_{sp} = V_{sp}^{(n)} - V_{sp}^{(n-1)} \]  
(48)

\[ \Delta V_{sh} = V_{sh}^{(n)} - V_{sh}^{(n-1)} \]  
(49)

The maximum error of the model is determined:

\[ Err_{\text{max}} = \max \left\{ \Delta P, \Delta H, \Delta V_{pi}, \Delta V_{pj}, \Delta V_{hj}, \Delta V_{hk}, \Delta V_{sp}, \Delta V_{sh} \right\} \]  
(50)

The algorithm will be terminated when either the maximum error \( Err_{\text{max}} \) is lower than a pre-specified tolerance or maximum number of iterations \( N_{\text{max}} \) is reached.

**Overall procedure**

Overall algorithm of the ALHN for the problem is as follows:

1. Select sigmoid function for continuous neurons and updating step sizes for all neurons.
2. Initialize inputs and outputs of all neurons.
3. Set \( n = 1 \).
4. Calculate dynamics of neurons using (10)-(15).
5. Update inputs of neurons by using (16)-(21).
6. Calculate outputs of neurons using (22)-(27).
7. If \( Err_{\text{max}} > \varepsilon \) and \( n < N_{\text{max}} \), set \( n = n + 1 \) and return to Step 4. Otherwise, stop.

**4. NUMERICAL RESULTS**

The proposed ALHN is tested on a system from [6]. The maximum error is set to \( 10^{-3} \). The algorithm of the proposed method is coded in MATLAB and run on 1.1GHz Celeron 128MB of RAM PC.

The test system has one pure power generation unit, two co-generation units, and one pure heat production unit. Data from the test system can be described as followings:

\[ F_I(P_j) = 50P_j \]

\[ F_I(P_j,H_j) = 2650 + 14.5P_j + 0.0345P_j^2 + 4.2H_j + 0.031P_jH_j \]

\[ F_I(P_j,H_j) = 1250 + 36P_j + 0.0435P_j^2 + 0.6H_j + 0.027H_j^2 + 0.011P_jH_j \]

\[ F_I(H_j) = 23.4H_j \]

\[ 0 \leq P_j \leq 150 \text{ MW} \]

\[ 0 \leq H_j \leq 2695.2 \text{ MWth} \]

Output limits of heat and power for co-generation units 2 and 3 are given in Fig. 3 and 4. The system power and heat demands are 200 MW and 115 MWth respectively.

Results obtained from the proposed ALHN are compared to LR [6], GA [7], IACSA [8], EP [9], and IGA-MU [10] in Table 1. The total production cost from the proposed method is very close to that of LR [6], EP [9], and IGA-MU [10] and less expensive than GA [7] and IACSA [8]. This shows that the proposed method is very efficient for the CHPED problem since it is simple for implementation and has a satisfactory result.
The proposed method is also implemented on large scale CHPED problem. In this implementation, four systems are considered with 20, 40, 80 and 120 units based on the basis of the four unit system. To obtain the system with 20 units, the basic four units are duplicated five times for number of units and load demands.

Table 1. Total cost comparison with other methods for 4 unit system

<table>
<thead>
<tr>
<th>Method</th>
<th>$P_1$ (MW)</th>
<th>$P_2$ (MW)</th>
<th>$P_3$ (MW)</th>
<th>$H_2$ (MWh)</th>
<th>$H_3$ (MWh)</th>
<th>$H_4$ (MWh)</th>
<th>Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR [6]</td>
<td>0</td>
<td>160</td>
<td>40</td>
<td>40</td>
<td>75</td>
<td>0</td>
<td>9,257.10</td>
</tr>
<tr>
<td>GA [7]</td>
<td>0</td>
<td>159.23</td>
<td>39.94</td>
<td>40.77</td>
<td>75.06</td>
<td>0</td>
<td>9,267.20</td>
</tr>
<tr>
<td>IACSA [8]</td>
<td>0.08</td>
<td>150.93</td>
<td>48.84</td>
<td>49.00</td>
<td>65.79</td>
<td>0.37</td>
<td>9,452.20</td>
</tr>
<tr>
<td>EP [9]</td>
<td>0</td>
<td>160</td>
<td>40</td>
<td>40</td>
<td>75</td>
<td>0</td>
<td>9,257.10</td>
</tr>
<tr>
<td>IGA-MU [10]</td>
<td>0</td>
<td>160.00</td>
<td>40.00</td>
<td>39.99</td>
<td>75.00</td>
<td>0</td>
<td>9,257.08</td>
</tr>
<tr>
<td>ALHN</td>
<td>0</td>
<td>159.9994</td>
<td>40</td>
<td>39.993</td>
<td>75</td>
<td>0</td>
<td>9,257.05</td>
</tr>
</tbody>
</table>

Table 2. Total costs comparison for up to 120 units system

<table>
<thead>
<tr>
<th>No. of units</th>
<th>CGA [10] ($$$)</th>
<th>IGA-MU [10] ($$$)</th>
<th>ALHN ($$$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>47,984.048</td>
<td>46,285.375</td>
<td>46,285.409</td>
</tr>
<tr>
<td>40</td>
<td>99,643.170</td>
<td>92,570.764</td>
<td>92,570.775</td>
</tr>
<tr>
<td>80</td>
<td>207,066.240</td>
<td>185,145.060</td>
<td>185,141.509</td>
</tr>
<tr>
<td>120</td>
<td>321,895.820</td>
<td>277,815.840</td>
<td>277,712.791</td>
</tr>
</tbody>
</table>

Table 3. CPU times comparison for up to 120 units system

<table>
<thead>
<tr>
<th>No. of units</th>
<th>CGA [10] (s)</th>
<th>IGA-MU [10] (s)</th>
<th>ALHN (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>264.47</td>
<td>86.95</td>
<td>0.37</td>
</tr>
<tr>
<td>40</td>
<td>534.54</td>
<td>168.67</td>
<td>0.38</td>
</tr>
<tr>
<td>80</td>
<td>1,053.69</td>
<td>343.01</td>
<td>0.42</td>
</tr>
<tr>
<td>120</td>
<td>1,489.14</td>
<td>471.27</td>
<td>0.47</td>
</tr>
</tbody>
</table>

5. CONCLUSION

ALHN is effectively implemented to solve CHPED problem. The ALHN is augmented by Hopfield terms from Hopfield network and penalty factors from Lagrangian relaxation that contribute to improve the convergence of the network. The proposed network has been tested on various-sized systems and results are less expensive than CGA and IGA-MU, especially for large scale systems. Moreover, the computational times of the ALHN are much faster than CGA and IGA-MU and slightly increase with the system size, which is very favorable for large scale implementation of CHPED problem.

REFERENCES