Proceedings of the IASTED International Conference
POWER AND ENERGY SYSTEMS
May 13-15, 2002, Marina del Rey, California, USA

MODIFIED HOPFIELD METHOD TO ECONOMIC DISPATCH OF UNITS WITH MULTIPLE FUEL OPTIONS

P.Somasundaram, S.G.Bharathi Dasan, G. Sadasivam
K.Kuppusamy, R.P.Kumudini Devi
School of Electrical and Electronics Engineering
Anna University, Chennai – 600 025,
India

ABSTRACT

This paper presents the application of fast computation Hopfield neural network to economic dispatch (ED) of generators having piecewise quadratic cost functions. Traditionally a convex cost function for each generator is assumed. However, it is more realistic to represent the cost function as a piecewise quadratic function rather than single convex function. In this study, multiple intersecting cost functions are used for each unit. The modified Hopfield method employs a linear input-output model for neurons. Formulations for solving the ED problems are explored. This method determines the weight factors of the energy function by direct computation where as in the usual Hopfield methods weight factors are calculated by trial and error method. The solution to the ED problem is also obtained by direct computation. The effectiveness of this method is tested by applying it to a sample system. Computational results manifest that the method has a lot of excellent performances, and it is superior to other methods in many respects.

KEY WORDS
Artificial Intelligence Applications, Economic Dispatch, Hopfield model, piecewise quadratic cost function.

1.INTRODUCTION

In power system operation and planning, economic dispatch is one of the most important criteria. Traditionally, the ED problems were solved with each generating unit having a single cost function. However, certain practical thermal units use different fuels like coal, natural gas and oil. This multiple fuel options lead to piecewise quadratic cost functions.

Lin and Viviani [1] used the lagrangian function for solving the ED problem with segmented piecewise quadratic cost functions. Park et al. [2] presented the solution to the ED problem with piecewise quadratic cost functions by using Hopfield neural network. Recently, Lee et al. [3] proposed solutions to the ED problem using adaptive Hopfield neural networks. The authors have compared the solution with those of the numerical approach and the conventional Hopfield neural network approach [2]. It has been shown that in the adaptive neural network approach the number of iterations required for converging to the optimum is half of the conventional Hopfield neural network method.

The Hopfield methods discussed above apply the iterative procedures thus require a large quantity of computation time. On the other hand Su and Chion [4] proposed the fast computation Hopfield neural network method to solve ED problems with each generating unit having a single cost function and the method is extended to solve the ED problem with transmission losses [5]. It does not include any iterative procedure and therefore the computational efforts were greatly reduced.

This paper investigates the application of the fast computation Hopfield neural network method to solve the ED problem with multiple fuel options for each generator. To show the effectiveness and validity of this method, it is implemented on a sample system and the results are compared with those obtained in the conventional Hopfield neural network approach [2].

2. FORMULATION OF ED PROBLEM

The objective of ED is to determine the optimal loadings for all on-line dispatchable units, which minimizes the total fuel cost while satisfying a set of constraints. It can be formulated as follows.

Fuel cost: Traditionally, in the ED problem, the fuel cost of each generator is represented by a single quadratic function. Here, owing to multiple fuel options, this cost function is piecewise quadratic. Hence the ED problem with piecewise quadratic cost function is defined as:

$$\min \quad f = \sum_{j=1}^{n} F_j(P_j), \tag{1}$$

$$F_j(P_j) = \begin{cases} a_{j1} + b_{j1}P_j + c_{j1}P_j^2, \ \text{fuel 1}, \ P_{j,min} \le P_j \le P_{j1}, \\ a_{j2} + b_{j2}P_j + c_{j2}P_j^2, \ \text{fuel 2}, \ P_{j1} \le P_j \le P_{j2}, \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ a_{jm} + b_{jm}P_j + c_{jm}P_j^2, \text{fuel } m, \ P_{j(m-1)} \le P_j \le P_{j,max}. \end{cases}$$

where

a_{jm}, b_{jm}, c_{jm} : cost coefficients of the j^{th} generator for fuel – type m

$F_j(P_j)$: fuel cost of generator j

f : total fuel cost

P_j : power output of j^{th} generator

n : number of generators

Power-balance constraint:

$$\sum_{j=1}^{n} P_j - P_D - P_L = 0, \tag{2}$$

where P_D is the total load demand and P_L is the transmission loss.

Capacity-limits constraint:

The power output level of generator j, should be between its minimum $P_{j,min}$ and maximum $P_{j,max}$:

$$P_{j,min} \le P_j \le P_{j,max} \tag{3}$$

3. MAPPING OF THE ED INTO HOPFIELD NETWORK

The dynamic characteristic of each neuron can be described by the following differential equation [4]

$$dU_i/dt = \sum_j T_{ij}V_j + I_i \tag{4}$$

where

U_i : the input of neuron i

T_{ii} : the self connection conductance of neuron i

T_{ij} : the interconnection conductance from the output of neuron j to the input of neuron i

I_i : the external input to neuron i

V_j : the output of neuron j

The energy function of the continuous Hopfield model can be defined as

$$E = -(1/2)\sum_i \sum_j T_{ij}V_iV_j - \sum_i I_i V_i \tag{5}$$

To solve the ED problem using Hopfield method, an energy function including both power mismatch, P_m, and total fuel cost, F_i, is defined as follows:

$$E = (A/2)[(P_D + P_L) - \sum_i P_i]^2 + (B/2)\sum_i (a_i + b_iP_i + c_iP_i^2)$$
$$= (A/2)P_m^2 + (B/2)F_i \tag{6}$$

where positive weighting factors A and B introduce the relative importance for their respective associated terms. With the application of the conventional Hopfield method to the ED problem, we can represent the power output value P_i using the output V_i of neuron i with a modified sigmoidal function described as follows [4]:

$$P_i = g_i(U_i)$$
$$= (1/2)(1 + \tanh(U_i/u_o))(P_{i\,max} - P_{i\,min}) + P_{i\,min} \tag{7}$$

where

u_o : the shape constant of the sigmoidal function

By comparing (6) with (5), we get

$$T_{ii} = - A - Bc_i \tag{8}$$

$$T_{ij} = - A \tag{9}$$

$$I_i = A(P_D + P_L) - Bb_i/2 \tag{10}$$

In proceeding the numerical computation, the following dynamic movement model will be used.

$$\Delta U_i = (\sum_j T_{ij}P_j + I_i)\,\Delta t \tag{11}$$

$$P_i = g_i(U_i) \tag{12}$$

where

ΔU_i : the differential variation of input U_i

Δt : the differential variation of time

The updating process using the above movement model for each neuron is repeated until the energy function converges to its minimum.

4. THE FAST COMPUTATION HOPFIELD METHOD

The input-output relationship of a neuron is expressed by a modified sigmoidal function in the conventional Hopfield method [2]. This type of input-output curve possesses saturation phenomena at its two ending regions. The saturation phenomena could result in incorrect dispatching levels for units during the iterating process. That is, a larger input variation might result in a smaller output change, or vice versa, as shown in Fig. 1, where

$$|\Delta U_i'| > |\Delta U_i| \text{ while } |\Delta P_i'| < |\Delta P_i| \tag{13}$$

On the other hand, the shape constant u_o of the sigmoidal function described in (7) affects the changing rate of the neuron's output P_i, with respect to the change of input, U_i. Too large a value of u_o will result in a much slow converging speed in the computation process. Conversely, too small a value of u_o will cause the function to behave as a two-value function, which often renders the neuron's output to be either at its upper limit or lower limit. Therefore, the selection of u_o is sophisticated.

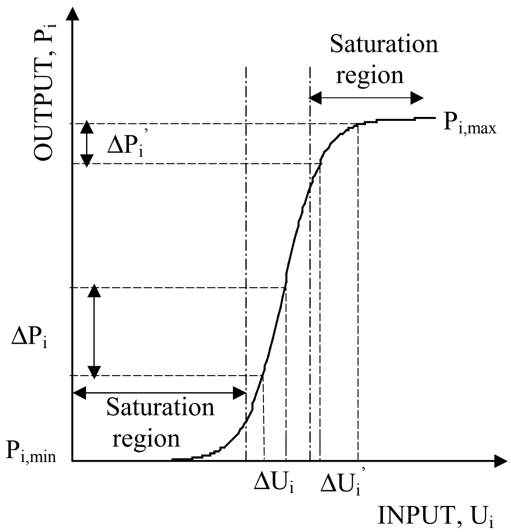

Fig.1 Modified sigmoid function

To avoid the problems resulting from curve saturation and improper selection of u_o, a novel linear model is proposed to describe the input-output relationship for the neuron.

4.1 Linear neuron model and energy function

A linear input-output model, $h_i(U_i)$, for a neuron can be written as

$$P_i = hi\ (U_i)$$
$$=[(U_i - U_{min}) / (U_{max} - U_{min})].(P_{i\ max} - P_{i\ min}) + P_{i\ min}$$
$$\equiv K_{1i}U_i + K_{2i} \quad \forall\ U_{min} \le U_i \le U_{max} \quad (14)$$

where both K_{1i} and K_{2i} are constants, and

$$K_{1i} = (P_{i\ max} - P_{i\ min})/(U_{max} - U_{min})$$
$$K_{2i} = -K_{1i}U_{min} + P_{i\ min}$$

and

$$P_i = h_i(U_i) = P_{i\ max} \quad \forall\ U_i > U_{max} \quad (15)$$
$$P_i = h_i(U_i) = P_{i\ min} \quad \forall\ U_i < U_{min} \quad (16)$$

The linear input-output relationship of neuron i is shown in Fig. 2. It is a piecewise continuous and monotonously increasing function, i.e., $h_i'(U_i) \ge 0$.

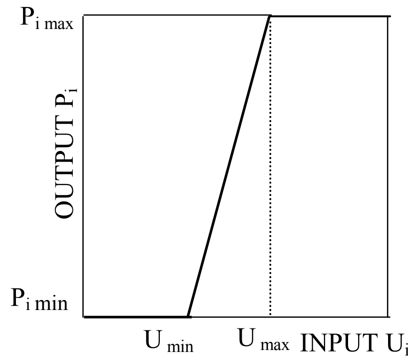

Fig.2 Linear input-output function

4.2 Computational expressions

Traditionally, the Hopfield method applies the sigmoidal function to describe the neuron's input-output characteristics and utilizes the iterative procedures to solve the problems. The proposed method employs the linear input-output model instead of the sigmoidal function, and the results are obtained by straightforward computation instead of iterative procedures. Moreover, power mismatch, P_m, can be preassigned to any small value such that the dynamic equation of a neuron has the merit that it is not related to any other neurons. Consequently, each neuron's dynamic performance can be simply described using a first-order linear ordinary differential equation. The expressions for directly computing the solutions are developed below.

The dynamic equation of a neuron is given as

$$dU_i/dt = \sum_j T_{ij}P_j + I_i \quad (17)$$

Substituting (8), (9), and (10) into (17), the dynamic equation becomes

$$dU_i/dt = AP_m - (B/2)(dF_i/dP_i) \quad (18)$$

The first term on the right side of the above equation bears no relation to the power output of unit i. However, the second term is related to the incremental cost associated with unit i. Hence, the dynamic performances of the neurons will bring about such a dispatching criterion that the units with lower incremental cost have a priority of further dispatching over the units with higher incremental cost.

Then, substituting (14) into (18), the dynamic equation becomes

$$dU_i/dt = AP_m - (B/2)[b_i + 2c_i(K_{1i}U_i + K_{2i})]$$
$$\equiv K_{3i}U_i + K_{4i} \quad (19)$$

where

$K_{3i} = -Bc_i(P_{i\,max} - P_{i\,min})/(U_{max} - U_{min})$

$K_{4i} = AP_m - (B/2)b_i - Bc_iK_{2i}$

Here, K_{3i} has relation to decaying speed, and its value is negative.

Solving (19), the neuron's input function, $U_i(t)$, is obtained as:

$$U_i(t) = [U_i(0) + (K_{4i}/K_{3i})]e^{K_{3i}t} + (-K_{4i}/K_{3i}) \quad (20)$$

From (14), the neuron's output function, $P_i(t)$, is obtained as:

$$P_i(t) = \{K_{1i}U_i(0) + K_{2i} - [(2(A/B)P_m - b_i)/(2c_i)]\}e^{K_{3i}t} + [(2(A/B)P_m - b_i)/(2c_i)]$$
$$= \{K_{1i}U_i(0) + K_{2i} - [(2K_{AB}P_m - b_i)/(2c_i)]\}e^{K_{3i}t} + [(2K_{AB}P_m - b_i)/(2c_i)] \quad (21)$$

where

$K_{AB} = A/B$

It is especially worth noting that two weighting factors A and B of the energy function used in the conventional model are now replaced by the factor K_{AB}. Therefore, difficulty of selecting the weighting factors is naturally avoided for the proposed model. Because $K_{3i} < 0$, the exponential term on the right side of the above equation is of transient existence. This term decays exponentially and finally becomes very small. Eventually, setting $t = \infty$ for (21), we have,

$$P_i(\infty) = (2K_{AB}P_m - b_i)/(2c_i) \quad (22)$$

Here $P_i(\infty)$ represents the optimal generation level for unit i, which is the solution we want. Also, from (22), it can be seen that $P_i(\infty)$ is not related to parameters (U_{max}-U_{min}) and $P_i(0)$, which is reasonable and comprehensible, and actually is another merit of the proposed model.

Applying (14) and (22) to (21), a more simple formula for the generation function is given as

$$P_i(t) = [P_i(0) - P_i(\infty)]e^{K_{3i}t} + P_i(\infty) \quad (23)$$

The power mismatch P_m, which is, defined as the load demand less the total generating power is expressed as

$$\sum_{i=1}^{n} P_i(\infty) = P_D - P_m \quad (24)$$

where

n : Number of the total units

Substituting (22) into (24) yields

$$\sum_{i=1}^{n} (2K_{AB}P_m - b_i)/(2c_i) = P_D - P_m \quad (25)$$

Rearranging the above equation, we obtain

$$P_m = \frac{P_D + (1/2)\sum_{i=1}^{n}(b_i/c_i)}{K_{AB}\sum_{i=1}^{n}(1/c_i) + 1} \quad (26)$$

Appropriately selecting K_{AB}, we have,

$$K_{AB}\sum_{i=1}^{n}(1/c_i) >> 1 \quad (27)$$

Finally, a useful approximate formula for P_m can be written as

$$P_m \approx \frac{P_D + (1/2)\sum_{i=1}^{n}(b_i/c_i)}{K_{AB}\sum_{i=1}^{n}(1/c_i)} \quad (28)$$

The expressions (22), (23) and (26) together make the Hopfield model for ED problems a direct computation.

4.3 The solution steps

The main computational steps are as follows:

Step 1 : Assume P_m, U_{max} and $U_{min.}$

Step 2 : Initialise $P_i^{(k)}(\infty)$, k=0.

Step 3 : Select the cost coefficients corresponding to $P_i^{(k)}(\infty)$.

Step 4 : Calculate K_{3i} and K_{AB} using (19) and (26) respectively.

Step 5 : Let k=k+1. Compute generation level $P_i(\infty)$ using (22) for all non-dispatch units.

Step 6 : Check the unit generation constraints. If any units violate their generation limits, go to step 7; Otherwise, the solution is obtained, and go to step 9.

Step 7 :

a) For each violated unit, compute time t (For convenience, imagine t as the time. Actually, t is a dimensionless variable.), to reach its generating limit using (23). (Initially, a set of feasible power loading for all units is required.)

b) From (a), identify the unit which first reaches its generation limit (i.e., the one with the smallest t computed in (a)), and dispatch the identified unit to output its limiting power that it hits.

c) Exclude the identified unit mentioned in (b) from the system which consists of non-dispatch units.

d) Subtract the limiting power mentioned in (b) from the total demand to get new total demand.

Step 8 : Go to step 3.

Step 9 : Is $|P_i^{(k-1)}(\infty) - P_i^{(k)}(\infty)|$ < tolerance for all units?

If yes go to next step; otherwise, go to step 3.

Step 10 : Stop the computation.

5. SIMULATION RESULTS AND DISCUSSIONS

The fast computation Hopfield neural network method is applied to the ELD problem with nonconvex cost functions. In reference [2] this problem was solved by conventional Hopfield neural network method, which is an iterative method. In order to prove the effectiveness of the proposed fast computation method, the data used in the conventional Hopfield neural network method [2] have been used. The optimal power dispatch with system demands rising from 2400 MW to 2700 MW is shown in Table 1 and Table 2.

Table 1. Results using conventional Hopfield neural network method

S	U	2400 MW		2500 MW		2600 MW		2700 MW	
		F	G	F	G	F	G	F	G
1	1	1	192.7	2	206.1	2	215.3	2	224.5
	2	1	203.8	1	206.3	1	210.6	1	215.0
	3	1	259.1	1	265.7	1	278.9	3	291.8
	4	2	195.1	3	235.7	3	238.9	3	242.2
2	5	1	248.7	1	258.2	1	275.7	1	293.3
	6	3	234.2	3	235.9	3	239.1	3	242.2
	7	1	260.3	1	269.1	1	286.2	1	303.1
3	8	3	234.2	3	235.9	3	239.1	3	242.2
	9	1	324.7	1	331.2	1	343.5	1	355.7
	10	1	246.8	1	255.7	1	272.6	1	289.5
GT		2399.8		2499.8		2599.8		2699.7	
C		487.87		526.13		574.26		626.12	

S: subsystem
F: fuel
U: unit
G: Unit Generation (MW)
GT: Total Generation (MW)
C: Total cost ($)

The results of the conventional Hopfield method and fast computation Hopfield method are shown in Table 1 and Table 2 respectively. Comparing Table 1 with Table 2, the following results are observed. First, the fast computation Hopfield neural network method satisfies the total load exactly, but, the conventional Hopfield method had a maximum power mismatch of 0.3 for 2700 MW load. Second, the total cost obtained by the fast computation neural network method is nearly the same as the conventional Hopfield neural network method.

Table 2. Results using fast computation Hopfield neural network method

S	U	2400 MW		2500 MW		2600 MW		2700 MW	
		F	G	F	G	F	G	F	G
1	1	1	190.9	1	194.8	2	211.0	2	219.6
	2	1	203.0	1	205.0	1	208.4	1	212.3
	3	1	255.7	1	261.5	1	271.5	1	282.4
	4	3	233.5	3	234.9	3	237.4	3	240.0
2	5	1	244.4	1	252.5	1	266.1	1	281.0
	6	3	233.5	3	234.9	3	237.4	3	240.0
	7	1	241.8	1	249.2	1	261.5	1	275.1
3	8	3	233.5	3	234.9	3	237.4	3	240.0
	9	1	322.0	3	382.8	3	406.6	3	432.5
	10	1	241.7	1	249.5	1	262.7	1	277.1
GT		2400.0		2500.0		2600.0		2700.0	
C		485.48		532.86		579.15		628.69	

The simulation time of the conventional Hopfield method with IBM PC- 1GHz is more than one minute, while the simulation time of the fast computation Hopfield method is less than 25 seconds.

6. CONCLUSIONS

The application of fast computation Hopfield neural network to economic dispatch problem with multiple fuel options is presented. The proposed algorithm has been tested on a ten-unit system. In comparison with the conventional Hopfield neural network approach the cost obtained by the proposed algorithm are nearly the same. The proposed algorithm is direct, powerful and easy to implement. These features render it as the most suitable method for solving practical economic dispatch problems with multiple fuel options.

REFERENCES

Journal Papers:

[1] C.E.Lin and G.L.Viviani, Hierarchical economic dispatch for piecewise quadratic cost functions, *IEEE Trans. on PAS, PAS-103*(6), 1984, 1170-1175.

[2] J.H. Park, et al., Economic load dispatch for piecewise quadratic cost function using Hopfield neural network, *IEEE Trans. on Power Systems, 8*(3), 1993, 1030-1038.

[3] K.Y. Lee, et al., Adaptive Hopfield neural networks for economic load dispatch, *IEEE Trans. on power systems, 13*(2), 1998, 519-526.

[4] C.T. Su and G.J. Chiou, A fast computation Hopfield method to economic dispatch of power systems, *IEEE Trans. on Power Systems, 12*(4), 1997, 1759-1764.

[5] C.T. Su and C.T. Lin, New approach with a Hopfield modelling framework to economic dispatch, *IEEE Trans. on Power Systems, 15*(2), 2000, 541-545.