A Reinforcement Learning approach to Hydrothermal Scheduling Problem

P. G. Latha
School of Engineering
Cochin University of Science and Technology
Kochi

T.P.Imthias Ahamed (M2009)
Saudi Aramco Chair in Electrical Power, Department of Electrical Engineering, College of Engg, King Saud University
Riyadh

P.S.Sreejith
School of Engineering
Cochin University of Science and Technology
Kochi

Abstract— **This paper proposes a Reinforcement Learning (RL) approach to the optimal scheduling of hydrothermal power system. Hydrothermal scheduling is an essential task performed by power system managers to decide the coordinated operation of hydro and thermal plants for achieving minimum production cost for the given time period. The application of RL has been successfully tried for various power system scheduling and control problems. The present work is an attempt to explore whether RL can be applied to hydrothermal scheduling which is one of the most challenging problems in power system**

Keywords-Reinforcement Learning; hydrothermal scheduling; multistage decision making; epsilon greedy algorithm

I. Introduction

Hydrothermal scheduling is one of the most important daily activities for a utility. Here the objective is to minimize the cost of running the thermal units by utilizing a given amount of water subject to system demand, reserve and individual unit constraints. Several methods have been proposed in the literature to solve the problem. These include Nonlinear Programming (NLP) [1], Dynamic Programming (DP) [2], Lagrangian Relaxation (LR) [3], Tabu Search (TS) [4], Expert Systems [5], Artificial Neural Networks (ANN) [6], Genetic Algorithms (GA) [7], Particle Swarm Optimization (PSO) [8] etc. Among these methods, the numerical solution methods are insufficient in handling large and complex systems. Soft computing methods can handle the above but they cannot handle stochastic data in practical systems. DP method provides a good framework for the problem. However it suffers from the “curse of dimensionality”. This problem is effectively tackled in Reinforcement Learning [9, 10]. While a DP algorithm operates with the entire state space, RL algorithm only operates on parts of the state space which are relevant to the system operation. The power of RL lies in its ability to solve, near-optimally, complex and large-scale multistage decision making problems on which classical DP breaks down. RL has been used as a powerful tool for many applications which can be modeled as a multistage decision making problem. The modern science of RL has emerged from a synthesis of notions from four different fields: classical DP, AI, stochastic approximation, and function approximation. Reinforcement learning combines the fields of dynamic programming and supervised learning to yield powerful machine-learning systems. Reinforcement learning appeals to many researchers because of its generality.

The main challenge in the hydrothermal scheduling problem is that most of the proven methods need a precise mathematical model of the system for getting an optimal solution. But this is very difficult due to complexity of the hydroelectric chain, non convex cost functions of thermal plants, stochastistity of inflow to hydel plants and cost of thermal plants etc. RL is a method of learning through interactions with environment. The main advantage of this approach is that it does not require a precise mathematical formulation. It can learn either by interacting with the environment or with the simulation model. Also unlike other methods like soft computing, the computational efforts for learning the dispatch for all possible load demands is almost same as the effort required to learn for one particular load demand. Although RL has been applied successfully to solve power system problems like Unit Commitment Problem (UCP), Economic Load Dispatch (ELD), Automatic Generation Control (AGC) etc., this method has not been investigated so far to solve the Hydrothermal Scheduling Problem. In this paper a short term hydrothermal scheduling (STHTS) problem is framed as a multistage decision making problem and solved using RL algorithm.

The rest of the paper is organized as follows. Section II presents the mathematical formulation of short term hydrothermal scheduling problem. A detailed description of RL technique, various concepts in RL and its algorithm is given in section III. The solution technique to tackle STHTS problem is discussed in section IV. Section V provides and analyses the results of a simple 2 plant model test system. Finally section VI presents the main conclusions of the paper.

II. Problem Formulation

A. *Objective function*

Hydrothermal scheduling is an optimization of a problem with the objective of minimizing the cost function which can be written as

$$\min J = \sum_{t=1}^{T}\sum_{i=1}^{N} F_i\,(P_s(i,t))$$

where T is the number of operating periods, N is the number of thermal plants, C is the composite cost function, $P_S(i, t)$ is loading of i^{th} thermal unit at time t and a_i, b_i, c_i are thermal generation cost coefficients.

B. Constraints

The constraints to be satisfied in this problem are as follows:

1) Load generation balance

$$\sum_{i=1}^{N} P_s(i,t) + \sum_{j=1}^{M} P_h(j,t) = P_L(t)$$

where M is the number of hydropower stations, $P_h(j, t)$ is loading of j^{th} hydro plant unit at time t and $P_L(t)$ is the load demand at time t.

2) Thermal plant loading limits

where P_{imin} & P_{imax} are the minimum and maximum power output of thermal unit i.

3) Hydro plant loading limits

where P_{jmin} & P_{jmax} are the minimum and maximum power output of hydro unit j.

4) Reservoir level limits

where V(j, t) is the storage of reservoir j at time t and V_{jmin}& V_{jmax} are the minimum and maximum storages of reservoir j

5) Reservoir discharge limits

where Q (j, t) is the discharge of reservoir j at time t and Q_{jmin} & Q_{jmax} are the minimum and maximum discharges of reservoir j

6) Water balance equation (2)

$$V(j,t) = V(j,t\,I\,1) + n_t\left(r(j,t)\,I\,Q(j,t)\,I\,S(j,t)\right) \qquad (7)$$

where r (j, t) & S (j, t) are the inflow and spillage for reservoir j during time t. Also n_t is the length of time slot in hour t

7) Initial and final reservoir limits

$$V(j,0) = V_0$$
$$V(j,T) = V_T \qquad (8)$$

where V_0 is the initial storage and V_T is the final storage of reservoir j during the scheduling period.

8) Water use rate characteristic of hydro plant

(9)

where d_j, g_j & h_j are the water use rate coefficients of hydro plant j

III. Reinforcement Learning

Reinforcement Learning (RL) is learning by interacting with an environment. An RL agent learns from the consequences of its actions, rather than from being explicitly taught and it selects its actions on basis of its past experiences (exploitation) and also by new choices (exploration), which is essentially trial and error. The reinforcement signal that the RL agent receives is a numerical reward which encodes the success of an action's outcome and the agent seeks to learn to select actions that maximize the accumulated reward over time. RL is generally used to solve multistage decision making problems.

Multistage decision making problems are modeled as Markov decision processes (MDPs) named after Andrey Markov. MDPs provide a mathematical framework for modeling decision-making in situations where outcomes are partly random and partly under the control of a decision maker. MDPs are useful for studying a wide range of optimization problems solved via Dynamic Programming and Reinforcement Learning. MDPs were known at least as early as the 1950s. Today they are used in a variety of areas including robotics, automated control, economics, manufacturing etc.

Now we explain how RL can be used to solve an MDP. There are various types of MDP. Here to introduce RL, an N stage decision making problem is explained. Let the system be in state x_0 in stage 0. If we take an action a_0, system will move to next state x_1. When the system moves from state x_0 to x_1, it will incur a cost $g(x_0, a_0, x_1)$. In general, in the k^{th} stage, if we take an action a_k, system will move from x_k to x_{k+1} and will incur a cost $g(x_k, a_k, x_{k+1})$. For an N stage MDP, system will reach an absorption state in the Nth stage. The total cost incurred when we start from x_0 and reach final state is

$$\sum_{k=0}^{N|1} \gamma^k g(x_k, a_k, x_{k+1})$$

where Υ is the discount factor. Υ is chosen a value between 0 and 1 depending on the significance of future costs.

MDP is the problem of finding actions $a_0, a_1, a_2 .., a_{N-1}$ such that the total expected cost C is minimized

$$C = E\left[\sum_{k=0}^{N|1} \gamma^k g(x_k, a_k, x_{k+1})\right] \qquad (10)$$

In general, the cost $g(x_k, a_k, x_{k+1})$ could be a random variable. When the system is in state x, we will take an action a based on some "policy". Usually, policy is denoted by $\pi(\,)$. Thus if we are following a policy $\pi(\,)$, the action taken in state x is $\pi(x)$. We can think of π as a mapping

$$\pi: X \mid A$$

where X is the state space and A is the action space. There are several algorithms to find optimal policy. Here, we explain one algorithm, the Q learning algorithm.

Q learning algorithm involves learning the Q values. Q value for a state action pair (x, a) is defined as the total expected cost, if we start from state x, take an action a, thereafter follow the optimal policy, π^*.i.e.,

where $x_0 = x$, $a_0 = a$, $a_k = \pi^*(x_k)$ *for* $k=1,2,3 \ldots. N-1$

Let the Q values for all possible actions $\{a_i : i=1,2,\ldots.m\}$ in state x *i.e.,* $Q(x, a_1), Q(x, a_2), \ldots\ldots\ldots\ldots Q(x, a_m)$, be known. Then if $Q(x, a^*) < Q(x, a_i)$ for all $a_i \neq a^*$, we say a^* is the optimal action in state x. Mathematically, we write,

Similarly, if the Q values for all state action pairs are known, then we can find the optimal policy or best action in any state x, using the equation

Thus, by learning Q–values for different possible state – action pairs, we can find the optimal actions. Therefore, to learn the optimal actions, we have to first learn the Q-values. To learn the Q-values, we proceed as follows. We start with an initial guess for $Q(x_k, a_k)$ which we denote as $Q_0(x_k, a_k)$. At each time instant k the system is in state x_k, we take an action a_k based on the current estimate of $Q^*(x_k, a_k)$, $Q^n(x_k, a_k)$. Based on the action chosen and dynamics of the system, we reach a new state x_{k+1} and incur a cost of $g(x_k, a_k, x_{k+1})$. Using this data, we update the Q value for the current state action pair using the following equation

(14)

where α the learning index, indicates how much the Q values are modified in each step. Similarly, Q values of the state- action pair corresponding to all the states visited are updated. After N stages, system will reach the terminal state; then we say an episode is over. In general, N can be a random number. To learn the Q values corresponding to all the relevant state-action pairs, we start the iteration from a new initial state and update the Q values of all states visited till the system reach the terminal state. In this manner a large number of episodes are repeated. An important issue in RL is how to select the actions during the learning phase. It may be noted that if we know the optimal Q values, we can find optimal actions using (12). However, in the initial part of the algorithm, the optimal Q values are unknown; but, we have an estimate of Q values $Q^n(x_k, a_k)$. The best action with respect to the estimate of the Q values is termed as the greedy action. Greedy action is given by the equation

$$a_g = \text{argmin}_{a_i} Q^n(x, a_i) \qquad (15)$$

One straight forward approach used in many RL applications is known as the ε-greedy algorithm. In ε greedy algorithm, on reaching any state, greedy action is selected with probability 1- ε and any other random action with probability ε. Now the issue is how to choose ε. In the initial part of the algorithm, ε is chosen close to 1 and as algorithm proceeds, ε is reduced to zero. There are various other sophisticated methods to judiciously balance the exploration and exploitation in RL literature [9-11].

The ε- greedy algorithm for Q – Learning

```
Initialize Q (x, a) to 0 for all (x, a) pairs
Initialize max. episodes, α, ε, and $x_g$
For i=1: max. episodes
  k=1
  x=1
  While $x \neq x_g$
    Find greedy action using (15)
    a=$a_g$ with probability 1-ε
    a=random action with probability ε
    Obtain the new state $x_{new}$ based on the current action
    Obtain the cost corresponding to the new transition
    $g(x_k, a_k, x_{k+1})$
    Update Q value corresponding to the state action pair
    (x,a) using (14)
    x=$x_{new}$
    k=k+1
  end
  Update ε based on the cooling schedule
  No. of steps (i) =k
  Estimate of expected cost ( i) = min(Q(i, a))
End
```

The above learning procedure is repeated a large number of times. During the learning process, Q values of the state-action pairs will be modified. In the initial phase of learning, the estimated Q values $Q^n(x_k,a_k)$ may not be closer to the optimal value $Q^*(x_k, a_k)$. As the learning progresses, the difference between the successive updates of Q values will become smaller and the estimates approach to the optimum i.e. the estimated Q values will converge to the true Q values. Once the optimal Q values are reached, the best action will be the greedy action at each stage k.

After completing the learning phase, the optimal policy has to be retrieved. Following is the algorithm for the retrieving phase.

```
Read the Q values
Get the initial state $x_0$
For k=0 to N-1
Do
  Find the greedy action using (16)
  Find the new state $x_{k+1}$
end Do
```

IV. RL Algorithm For Hydrothermal Scheduling

RL has been successfully applied for control and scheduling applications [12-18]. But its applicability for solving hydrothermal scheduling has not been explored so far. In this paper we formulated the short term hydrothermal scheduling (STHTS) problem as an MDP and solved using RL algorithm. Here we consider a simple test system with a single hydroplant P_h operated in conjunction with a thermal plant P_S serving a single series of load P_L.

To use RL for solving STHTS problem, the first step is to frame it as an MDP. Here the number of stages is taken as the number of time slots in the scheduling period. After stating the problem as an MDP, the next step is to define the state space X, action space A and a reinforcement function g. Then a simulation model for the problem has to be derived.

For this, the reservoir volume between the end limits is discretised in suitable steps. This constitutes the state space of the problem where the end states of this MDP are the initial and final reservoir volume specified for the scheduling horizon. The state at any stage is a particular volume in the hydro plant reservoir. The action space is the set of all volume states between the minimum & maximum limit of the reservoir.

A single plant hydrothermal system has the following RL framework for the short term scheduling problem

$$X=x_1 \cup x_2 \cup \ldots\ldots x_T$$

where the state at any stage $x_k \in X$ is any discrete volume. The decision taken at each stage is by selecting an action from the action space A where

$$A= A_1 \cup A_2 \cup \ldots\ldots A_T \text{ such that}$$

$A_k = \{a_k : V_{min} \le a_k \le V_{max}\}$

Next we require a simulation model that gives the next state x_{k+1} given a current state x_k and action at that stage a_k. In the case of STHTS problem, the model is very simple. The initial state and the goal state are the reservoir volume limits specified for the beginning and end of the scheduling period. At any stage k any action a_k is taken as explained in the ε-greedy algorithm and the learning agent reaches a new state x_{k+1}. Selection of x_{k+1} at stage k decides the hydro plant loading $P_h(k)$ from which the thermal power $P_s(k)$ can be calculated as

The cost incurred for transition from state x_k to x_{k+1} denoted as $g(x_k, a_k, x_{k+1})$ is the production cost C(k) of the thermal plant $P_s(k)$. The action a_k from an action space A_k can be judiciously selected such that the constraints are met. For example, the selection of a non feasible value of volume state can be discouraged by assigning a very high value of g. Here each stage k corresponds to a load $P_L(k)$. Using epsilon greedy algorithm as explained in section III, the optimal schedule can be learned for all loads. The total production cost of the entire scheduling period can be calculated as

TABLE III. SYSTEM LOAD DATA

Period	P_L (MW)
1	600
2	1000
3	900
4	500
5	400
6	300

In this framework, the objective of the problem is to find sequence of actions $a_0, a_1, a_2, \ldots\ldots\ldots a_{N-1}$ such that total cost C is minimized.

For selecting an action from the action set, we use ε-greedy method. During the learning phase, we learn the Q-values for all possible combinations of state-action pair. The optimal schedule is then found by following the algorithmic steps explained in section III.

V. SIMULATION RESULTS

The proposed algorithm is implemented in Matlab 7.10.0 and is tested for two plant hydrothermal system given in [19]. Both the plants are single unit plants. The details of the two plants are shown in table I & table II. The thermal cost coefficients are given in terms of ₹ which is a fictitious monitory unit. Also it is assumed that, hydro unit considered is a constant head plant with a constant natural inflow of 1000 acre-ft/hr. Table III shows the load data for a 24 hour day with individual periods taken as 4 hr each.

TABLE 1. HYDROPLANT CHARACTERISTICS

P_{min} (MW)	P_{max} (MW)	d (acre-ft/hr)	g (acre-ft/hr-MW)	h (acre-ft/hr-MW^2)	V_{min} (acre-ft)	V_{max} (acre-ft)
0	200	260	10	0	6000	18000

TABLE II. THERMAL PLANT CHARACTERISTICS

P_{min} (MW)	P_{max} (MW)	a (₹/MW^2-hr)	b (₹/MW-hr)	c (₹/hr)
2000	1200	700	4.8	1/2000

To run the algorithm, we have to choose proper values for the learning parameters. This is done by trial and error. The learning parameter ε accounts for the rate of exploration and exploitation needed. Here a value of 0.5 is selected initially providing sufficient exploration of the search space and is decreased in steps successively. A value of α as 0.1 showed sufficiently good convergence. Since the cost of future stages has the same implication as the cost of the current stage, value of Υ is taken as 1.

The final minimum cost trajectory for the storage volume is plotted in Figure 1. The optimal cost is 81738.46 ₹. The solution is exactly same as the one using Dynamic Programming. This shows the success of RL for deterministic data. DP cannot be used in random environment whereas RL can effectively tackle this issue. Also, the thermal cost function in this case can be non convex in nature. The optimal path can be determined to a rather course grid of 2000 acre-feet by 4 hr steps in time and could easily be recomputed with finer increments.

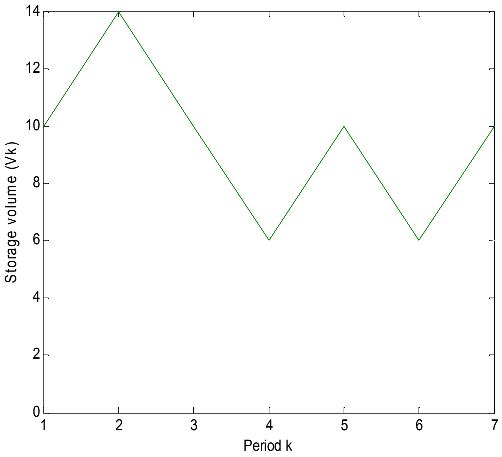

Figure.1 Final trajectory for the hydrothermal scheduling

.

VI. CONCLUSIONS

In this paper we have demonstrated the application of RL for solving one of the challenging problems in power system. Unlike most of the other methods, RL does not need a precise mathematical model of the problem. Another important feature of RL is that, it can effectively handle the stochastic cost functions associated with practical thermal units. The performance of the algorithm is good for the test system available in the literature.

The multiplant hydraulically coupled systems offer computational difficulties that make it difficult to use that type of system to illustrate the benefits of applying RL to this problem. So in this study, we have considered a simple test system. An interesting future line of study is to apply the developed framework for the scheduling of practical power systems. In practical systems, in addition to the stochastic cost behavior of thermal units, a substantial randomness can be observed in the hydro plant inflow, power purchase cost etc.. Hence for power deficient hydrothermal systems involving day ahead and real time power purchase, RL can be a promising solution strategy.

REFERRENCES

[1] H. Habibollahzadeh and J. A. Bubenko, “Application of Decomposition Techniques to Short Term Operation Planning of Hydro-Thermal Power System,” IEEE Transactions on PWRS, Vol. 1, No. 1, February 1986, pp. 41-47.

[2] J. Yang and N. Chen, “Short term hydrothermal coordination using multi-pass dynamic programming,” IEEE Trans. Power Syst., vol. 4, pp. 1050–1056, Aug. 1989.

[3] J. M. Ngundam, F. Kenfack and T. T. Tatietse, “Optimal Scheduling of Large-Scale Hydrothermal Power Systems Using the Lagrangian Relaxation Technique,” International Journal of Electrical Power and Energy Systems, Vol. 22, No. 4, 2000, pp. 237-245.

[4] X. M. Bai and S. M. Shahidehpour, “Hydro-Thermal Scheduling by Tabu Search and Decomposition Method,” IEEE Transactions on Power Systems, Vol. 11, No. 2, 1996, pp. 968-974.

[5] S. Li, S. M. Shahidehpour and C. Wang, “Promoting the Application of Expect Systems in Short-Term Unit Commitment,” IEEE Transactions on Power System, Vol. 3, No. 1, May 1993, pp. 286-292.

[6] M. Basu, “Hopfield neural networks for optimal scheduling of fixed head hydrothermal power systems,” Electric Power Systems Research 64 (2003) 11–15.

[7] Zoumas C. E., Bakirtzis A. G., Theocharis J. B., et al., "A genetic algorithm solution approach to the hydrothermal coordination problem," IEEE Transactions on Power System, Vol. 19, No. 2, pp. 1356-1364, 2004.

[8] B. H. Yu, X. H. Yuan and J. W. Wang, “Short-Term Hydro-Thermal Scheduling Using Particle Swarm Optimization Method,” Energy Conversion & Management, Vol. 48, No. 7, 2007, pp. 1902-1908.

[9] R. S. Sutton and A. G. Barto (1998) “Reinforcement Learning: An Introduction”, MIT Press, Cambridge, MA.

[10] D . P. Bertsekas and J. N. Tsitsikilis (1996) Neuro Dynamic Programming, Athena Scientific, Belmount MA.

[11] M. A. L. Thathachar and P. S. Sastry, “Networks of Learning Automata: Techniques for on line stochastic optimization, Kulwer Academic, Boston, 2003.

[12] Buijtenen. W, Scharm. G, “Adaptive fuzzy control of satellite attitude by reinforcement learning ,” IEEE Transactions on Fuzzy Systems, Vol.6, No.2, May 1998, pp.185-194.

[13] Y. Hasegawa, T. Fukuda and K. Shimojima, Self-scaling reinforcement learning for fuzzy logic controller-applications to motion control of two-link brachiation robot. IEEE Transactions on Industrial Electronics., Vol.46, No.6, Dec. 1999, pp. 1123–1131.

[14] T.P.Imthias Ahamed, P.S.Nagendra Rao and P.S.Sastry, “A Reinforcement Learning approach to Automatic Generation Control,” Electric Power Systems Research, 63 (2002): 9-26.

[15] Wang, Y.C. & Usher, J.M, “Application of reinforcement learning for agent-based production scheduling,” Engineering Applications of Artificial Intelligence,Vol. 18, No.1, 2005, pp. 73-82.

[16] T.P.Imthias Ahamed,. “A Reinforcement Learning Approach to Unit Commitment Problem,” Proceedings of National Power System Conference 2006.

[17] E. A. Jasmin, T. P. Imthias Ahamed, V. P. Jagathy Raj, “Reinforcement Learning approaches to Economic Dispatch Problem,” International Journal of Electrical Power and Energy Systems, Vol. 33, 2011, pp.836–845.

[18] T. P. Imthias Ahamed, S. Danish Maqbool & N.H. Malik (2011), “A Reinforcement Learning Approach to Demand Response,” Centenary Conference EE, IISc Bangalore, India.

[19] A.J.Wood, B.F.Woolenberg, “Power Generation and Control,” John Wiley Sons 2002.