Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 46 (2015) 202 – 209

Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

Reinforcement Learning in Adaptive Control of Power System Generation

Leo Raju[a*], Milton R S[a], Swetha Suresh[a], Sibi Sankar[a]

[a] *SSN College of Engineering,OMR,Chennai,603110,India.*

Abstract

Considering our depleting resources, efficient energy production and transmission is the need of the hour. This paper focuses on the concept of using Reinforcement Learning (RL) to control the power systems unit commitment and economic dispatch problem. The idea of reinforcement learning strives to present an ever optimal system even when there are load fluctuations. This is done by training the agent (system), thereby enriching its knowledge base which ensures that even without manual intervention all the available resources are used judiciously. Also the agent learns to reach long term objective of minimizing cost by autonomous optimization. A model free reinforcement learning method called, Q learning is used to find the cost at various loadings and is compared with the conventional priority list method and the performance improvement due to Q learning is proved.

Keywords: Unit commitment; Economic dispatch; Reinforcement Learning; Q Learning; Optimization

1. Introduction

Unit Commitment problem (UC) is a basic level in the scheduling operation of electrical power system. UC is a constrained optimization problem. It deals with scheduling a set of units to meet the forecasted load demand over a time period under different operational constraints[1]. The main objective is to reduce the total generation cost. Many

*Corresponding author .Tel: 91-44-27469700; fax: 91-44-27469772

Email address: leor@ssn.edu.in

doi:10.1016/j.procs.2015.02.012

Solutions to the UC problems are posed in the literature[3]. The conventional method is Priority List (PL) method, which is reasonably fast method[4], but here, the global solution is slightly different from the local optimized one. Genetic Algorithm (GA) is relatively successful in finding solution to UC problems[9]. GA method is further improved by combining with other methods to form hybrid methods[3,4,8]. Large problems use simulated annealing method for scalability[11,12]. But for large problems, Lagrangian Relaxation methods are the fittest ones[5]. In all these methods the concept of learning and learning through interaction are not emphasized. Reinforcement Learning (RL) is used for unit commitment problem[1,6,7,]. In this paper, the economic dispatch problem is solved with a model free Reinforcement Learning (RL) method, called Q Learning[1,2, 6]. Learning from interaction is a fundamental idea underlying the theories of learning. Reinforcement learning is a computational approach for automated, goal-directed learning and decision-making. It is better than other form of approaches by its emphasis on learning through direct interaction with its environment. The system is not just looking for the immediate reward but looks for maximizing the total expected reward by trial and error method. The system observes the environment and takes action to achieve its long term objectives of reducing the cost of power generation**.**

The paper is organised follows. In Section 2, unit commitment and economic dispatch problem is presented. Section 3 and 4 gives an explanatory discussion on Markov Decision Process and Reinforcement Learning. In Section 5, Q Learning is discussed in detail. Section 6 explains implementation of the Q Learning algorithm and Priority List method for the unit commitment and economic dispatch problem along with simulation and performance evaluation.

2. Unit commitment and economic dispatch

The optimum load dispatch problem deals with two kinds of problem. They are unit commitment and economic dispatch problems. The unit commitment is about selecting optimally out of the existing electrical resources to meet the load demand and providing margin of operating reserve over a specific time[4]. On-line economic dispatch problem deals with reducing the total cost of operation by proper distribution of the load among the generating units. It is a constraint optimization problem which considers various power generation constraints along with cost reduction when allotting the load to the generators. Economic load scheduling is about finding the different combination of generators, so as to minimize the fuel cost. Also the total generation must meet the load demand and losses at any instant.

3. Markov Decision Process

Markov Decision Process (MDP) is a way to model a sequential decision making under uncertainty. We formalize an MDP, considering discrete states and actions. The initial state is 's' and each state will have a reward 'r' associated with it. The transition function T (s|a, s') indicates the probability of transitioning from state 's' to s' when action 'a' is taken. A discount factor γ in the range 0 to1 is applied to future rewards[6]. This represents the notion that a current reward is more valuable than one in the future. If it is near zero, future rewards are almost ignored; a near one places great value on future reward. The reward from a policy is the sum of the discounted expected utility of each state visited by that policy. The policy that maximizes the total expected discounted reward is called as optimal policy. RL method works in the MDP environment. The state signal is said to possess Markov property when it retains relevant past information to predict the future. This property implies a stochastic process in which the probability of an event depends only on the immediately preceding event and not the past.

4. Reinforcement Learning

Reinforcement Learning is the most efficient method in giving solutions to sequential decision making problems. Reinforcement learning algorithm is used to model the agent's adaptation to a dynamically changing environment by performing actions in an MDP environment. The agents observe the environment and take an action. It gets a reward or punishment from the environment. The training information is used to evaluate actions (in terms of reward or punishment received from the environment) taken by the agent. The agent takes the next action to optimize the reward in the long run. After a number of interactions, with the enough learning, the agent finds the optimal policy to achieve long term objective. The algorithm stores the values of the state, action and reward at each step and uses this data in the analysis of the next step. Agent makes the interface between the environment and the algorithm. The agent selects actions and the environment responds by presenting new situations to the agent.

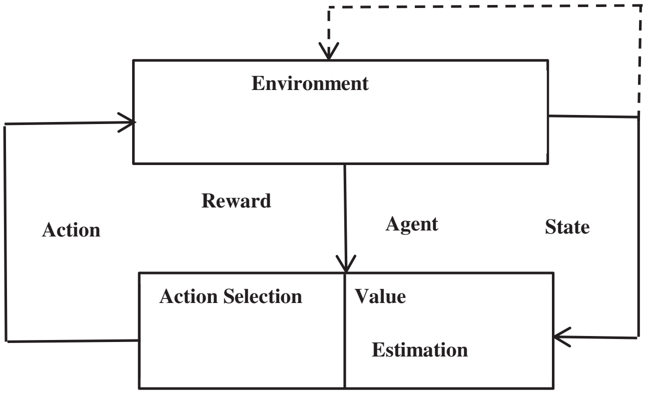


Fig.1 Reinforcement Learning

5. Q Learning

Q learning is a model-free reinforcement learning where the agent explores the environment and finds the next reward plus the best the agent can do from the next state. It does not require a model and it only needs to know what states exist and what actions are possible in each state. We assign each state an estimated value, called a Q value. When an agent visits a state and takes an action it receives a reward. It uses this reward to update the estimate of the value of that action in the long run. The agent visits the states infinitely often and the action values (Q values) are continuously updated till it becomes convergent. In the algorithm, γ is the discount factor and α, learning rate. The environmental states are represented as s(t) and possible actions as a(t) and the agent learns the value of each of those actions in each of those states. These values are called Q values or state action values. Here the agent starts by initializing all the Q values to 0, and goes around and explores the environment. After an action is taken in a state, the agent evaluates the state that it has led to. If it has led to an undesirable outcome or the agent is punished, then the Q value (or weight) of that action is reduced from that state so that other actions will have a greater Q value and be chosen in the next time the agent is in that state. Similarly, if the agent is rewarded for taking a particular action, the weight of that action for that state is increased, so the agent more likely to choose it again the next time when it is in that state. When the Q value is updated, the previous state-action combination is updated. So the agent can update its Q value only after it has seen the results.

The Q-Learning algorithm[2] goes as follows:

1. Set the discount factor, and rewards in matrix R.

2. Q values are initialised to zero in Q matrix.

3. for each episode:

   A random initial state is selected.

   Do while the goal state (convergent) hasn't been reached.

- Select one among all possible actions for the current state.
- Using this action, go to the next state.
- Get maximum Q value for this next state based on all possible actions.
- Compute: $Q(s,a) = Q(s,a) + \alpha[r + \gamma \max Q(s',a') - Q(s,a)]$ (1)
- $s \leftarrow s'$
- Set the next state as the current state.

End Do

End For

After sufficient iteration, $Q(s,a)$ converges to the Q value of the particular state action pair.

6. Simulation result

Three generating units PG1, PG2 and PG3 are considered with operating limits as follows:

$$150\text{MW} \le \text{PG1} \le 600\text{MW}$$
$$100\text{MW} \le \text{PG2} \le 400\text{MW}$$
$$50\text{MW} \le \text{PG3} \le 200\text{MW}$$

Production Cost Function of each generator is given below with coefficients of cost functions

$$F_1(P_1) = 0.00128P_1^2 + 6.48P_1 + 45 \quad (2)$$

$$F_2(P_2) = 0.00194P_2^2 + 7.85P_2 + 310 \quad (3)$$

$$F_3(P_3) = 0.00482P_3^2 + 7.97P_3 + 78 \quad (4)$$

P1, P2 and P3 are power required by the load.
The load to be shared is varied over time. The Startup costs for the machines are taken as Rs.200, Rs.190, and Rs.150 and the shut down costs for the machines are taken as Rs.100, Rs.180, and Rs.50

λ is calculated by using the following equation (5)

$$\lambda = \{P_D + \sum_{i=0}^{n}(b_i / 2a_i)\} / \{\sum_{i=0}^{n} 2a_i\} \quad (5)$$

P_D is power delivered and a_i , b_i are coeficients of cost function

Substituting values in the above equation, λ value is found to be 8.284X10^2 Rs / MWh

Power generated (P_{Gi}) is calculated using incremental cost basis

$$P_{Gi} = (\lambda - b_i) / 2a_i, \quad i = 1,2,3...N$$

If the computed P_{Gi} satisfies the operating limits, i.e,

$$P_{Gi,\min} \le P_{Gi} \le P_{Gi,\max}, \; i = 1,2,3...N$$

then optimum solution is obtained .

If P_{Gi} violates the operating limits then power generation is fixed at respective limits.

If $P_{Gi} \leq P_{Gi,\min}$, then $P_{Gi} = P_{Gi,\min}$

If $P_{Gi} \geq P_{Gi,\min}$, then $P_{Gi} = P_{Gi,\max}$

Thus the value of P_{Gi} is fixed.

For the above system a load value of 850MW is considered and conventional Lagrange multiplier method is applied to obtain the economic load dispatch.

λ is calculated by using the equation (5) and the power delivered from the three generation units are calculated as
P1= 704.6 MW
P2=111.8MW
P3=32.6MW

The solution satisfies the power balance equation. However units 1 and 3 are not within the limits. So the values are set to be:
P1=600MW
P2=200MW
P3=50MW

$$dF_i(P_{Gi}) / dP_{Gi} = \lambda \text{ for } \leq P_{Gi} \leq P_{Gi,\max} \text{ , } \leq \lambda \text{ for } P_{Gi} = P_{Gi,\max} \text{ and } \geq \lambda \text{ for } P_{Gi} = P_{Gi,\min}$$

λ includes the incremental cost of unit 2 since it is within the limit.

λ at PG_2= 200 is 8.626 X 10^2 Rs/MWh

λ at PG_1= 600 is 8.016 X 10^2 Rs/MWh

λ at PG_3= 50 is 8.452 X 10^2 Rs/MWh

Table 1 shows the economic load dispatch of the above mentioned loads, neglecting the transmission line losses.

Table 1: Lambda iteration method neglecting losses for three unit system

Unit no.	Loading Limits		Fuel Cost Parameter			Start up cost	Shutdown cost
	Min	Max	a_i	b_i	c_i	(manufacturer details)	
1	150	600	0.00128	6.48	459	200	100
2	100	400	0.00194	7.85	310	190	180
3	50	200	0.00482	7.97	78	150	50

Now the incremental cost for unit 1 is less than λ, so unit 1 must be at its maximum. But the incremental cost of unit 3 is not less than λ and so it is not in its maximum value. To calculate economic dispatch, we fix unit 1 at 600MW and split 250MW between unit 2 and 3. By identical splitting we get P_1 = 600 MW, P2 =187.1MW, P_3 = 62.9MW.

Cost for each unit = production cost + startup cost

From equation 2, 3, 4 and the startup costs, the total incurred cost in PL method, which is sum of production cost and startup for the three generators, is calculated as follows

Unit1 = 4807.8 + 200 = 5007.8

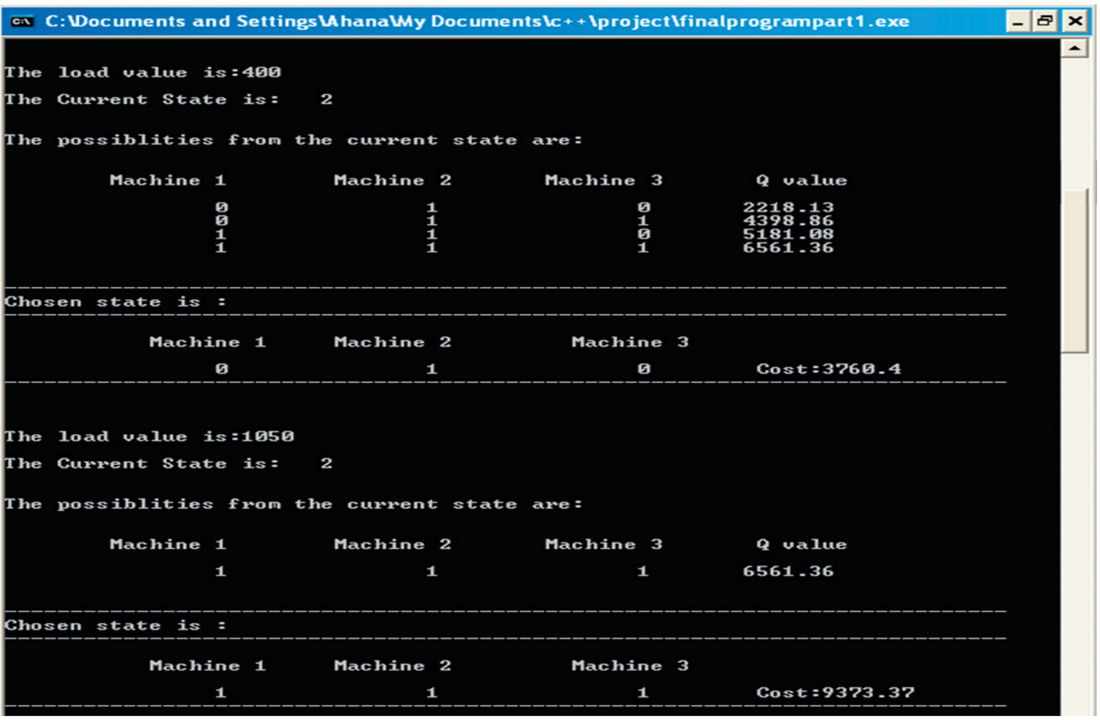
Unit2 = 1846.6 + 190 = 2036.6

Unit3 = 598.38 + 150 = 748.38.

Thus the total incurred cost by Priority List (PL) method, is calculated to be Rs 7792.98 /MWh. In Table 2 the generator switch is considered as closed when it is "ON" and open when it is "OFF". In the computer output screen shot, shown in Fig.2 machine state "1" is taken as "ON" and "0" is taken as "OFF"

Table 2: Operational costs with Lambda Iteration method

Sl. No.	Unit I	Unit II	Unit III	P_{G1}	P_{G2}	P_{G3}	Solution	Cost
1	open	open	open	-	-	-	Infeasible	-
2	open	open	closed	-	-	-	Infeasible	-
3	open	close	open	-	-	-	Infeasible	-
4	open	close	close	-	-	-	Infeasible	-
5	close	open	open	-	-	-	Infeasible	-
6	close	open	close	-	-	-	Infeasible	-
7	close	close	open	600	250	-	Feasible	7591.55
8	close	close	close	600	187.1	62.9	Feasible	7792.78


C:\Documents and Settings\Ahana\My Documents\c++\project\finalprogrampart1.exe

```
The load value is:400
The Current State is:   2

The possiblities from the current state are:

        Machine 1          Machine 2         Machine 3         Q value

                0                  1                 0        2218.13
                0                  1                 1        4398.86
                1                  1                 0        5181.08
                1                  1                 1        6561.36

----------------------------------------------------------------------------
Chosen state is :
----------------------------------------------------------------------------
           Machine 1       Machine 2           Machine 3

                0                  1                 0          Cost:3760.4
----------------------------------------------------------------------------


The load value is:1050
The Current State is:   2

The possiblities from the current state are:

        Machine 1          Machine 2         Machine 3         Q value
                1                  1                 1        6561.36

----------------------------------------------------------------------------
Chosen state is :
----------------------------------------------------------------------------
           Machine 1       Machine 2           Machine 3
                1                  1                 1          Cost:9373.37
----------------------------------------------------------------------------
```

Fig.2. Output for load values of 400MW and 1050MW

Then cost is calculated by Q Learning method by reading the initial status of the different generating units from the unit data and the different possible states and actions are identified. In the beginning, Q values are initialized to zero. At every time step, the unit action should be in such a way to satisfy the load requirement. Therefore, using the forecasted load profile and the unit generation

Table 3 Comparison between PL and RL method costs for different loads

LOAD(MW)	COST-RL in Rs	COST -PL in Rs
850	7591.55	7792.83
200	2057.6	2107.6
400	3760.4	4836.75
1050	9373.37	9023.37
975	8384.36	8346.45
1000	1214.4	1264.4
850	7401.55	7602.83
980	8380.94	8391.1
140	1547.02	1597.02
230	2218.13	2218.13
600	5016.64	6363.15
TOTAL	56946	59543.6

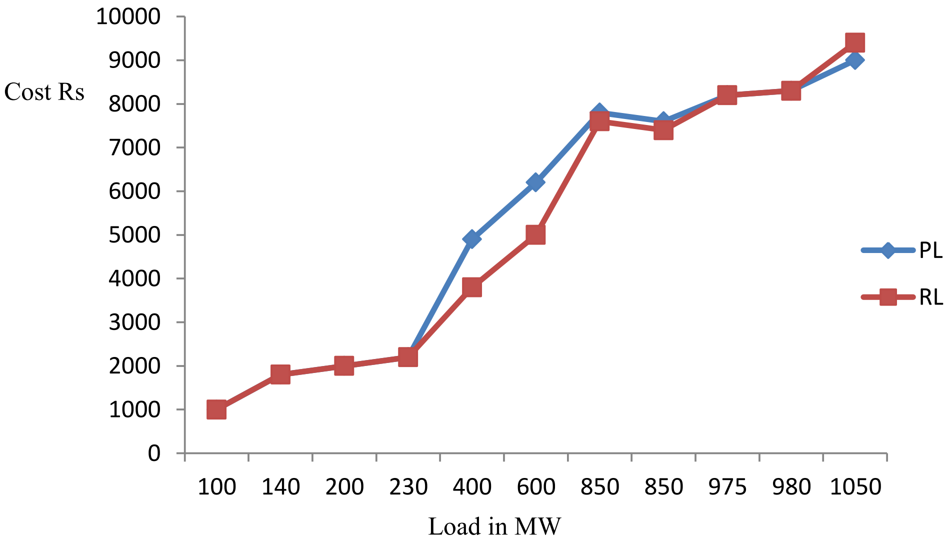


Fig. 3. Plot of Load vs. Cost for PL and RL method

constraints, the set of feasible actions are identified for each state. Using the greedy strategy one of the possible action is taken and so the state transition occurs. The net cost of generation is taken as a reward. Using the reward, estimated Q value is updated at each of the stages until the last stage using the equation (1). This constitutes one episode. In each episode the algorithm passes through all the stages. Then the algorithm is executed again from the beginning state. These episodes are repeated many times. After some time the above iteration converges to final Q value. The combination of load for which the value of Q is minimum, is the lowest cost combination among all possible combinations. The Implementation of Q Learning is done in C++ programming language. Q value and the cost for load values of 400MW and 1050MW are shown in Fig. 2. Q Learning takes some time to learn initially by trial and error and once it learns sufficiently, it functions optimally towards long term objectives of minimizing the cost. In Fig.3, the total cost by Q Learning method is compared with Priority List method for various loads given in

Table 3. Q learning method is proved to be more economical than PL method in the long run.

7. Conclusions

Thus this paper provides an optimal solution for the unit commitment and economic dispatch problems by using a model free Reinforcement Learning method, called Q learning. This method also enhances the adaptability of the system in dynamic environment. The system learns continuously towards long term objective of reducing the cost of power generation. It also enables management of unforeseen load. Q learning technique provides a wholesome procedure to effectively control a power system in the dynamic environment and thus reduce the cost considerably when compared to the conventional Priority List method. In future single agent reinforcement learning can be extended to Multi-Agent Reinforcement Learning to accommodate other type of renewable energy resources like solar and wind along with the thermal units.

References

1. Carla A Coronado, Marcelo R Figueroa, Claudio A Roa-Sepulveda. A Reinforcement Learning Solution for the Unit commitment . Problem. 47th International Universities Power Engineering conference, London, 2012.
2. Richard S. Sutton, Andrew G. Barto. *Reinforcement Learning: An Introduction.* London, MIT Press; 1998.
3. Sayeed Salam. Unit Commitment Solution Methods. *Proceedings of World Academy of Science, Engineering and Technology*, vol. 26; 2007, p. 2070-3740.
4. A Saravanan, Siddharth Das, Surbhi Sikri. A solution to the unit commitment problem—a review. *Frontiers in Energy,* June 2013, Volume 7, Issue 2, pp 223-236
5. S. Takriti, J.R. Birge. Using integer programming to Lagrangian based unit commitment solutions. *IEEE Transaction on Power Systems*, vol. 15; 2000, p. 151-156.
6. T. P. Imthias Ahamed, E. A. Jasmin, Essam A. Al-Ammar. Reinforcement Learning in Power System Scheduling and Control: A Unified Perspective. *IEEE Symposium on Computer and Informatics (ISCI 2011),* Malaysia; 2011.
7. E.A.Jasmin,T.P.Imthias Ahamed,V.P.Jagathy Raj.Reinforcement Learning approaches to Economic Dispatch problem. *International Journal of Electrica Powerl&Energy Systems,* Vol.33, no. 4, May 2011, p.836-845.
8. A. Bhardwaj. Unit Commitment in Electrical Power System - a Literature Review. *IEEE International Power Engineering and Optimization Conference*, Malaysia, June, 2012.
9. Swarup, K.S. Yamashiro, S. Unit commitment solution methodology using genetic algorithm *IEEE Transaction on Power Systems*, vol. 17; 2002, p. 87-91
10. A.G. Bakiritzis, D.E. Zoumas. Lamda of Lagrangian relaxation solution to unit commitment problem. *IEEE Proceeding on Generation, Transmission and Distribution*, vol. 147; 2000, pp. 131-136.
11. F. Zhuang, F. Galiana. Unit commitment by simulated annealing. *International Journal of Electrical Power & Energy Systems*, vol. 5; 1990, p. 311-318.
12. A.Y. Saber, T Senjyu, T Miyagi, N Urasaki, T. Absolute stochastic simulated annealing approach to Unit Commitment problem. *IEEE 13th International conference on Intelligent System Application to Power System*, Virginia, 2005.