2009 3rd International Conference on Power Electronics Systems and Applications

Cooperative Reinforcement Learning Algorithm to Distributed Power System based on Multi-Agent

La-mei GAO[1], Jun ZENG [1], Jie WU [1] , Min LI[1]

[1]Gao La-mei is with the College of Electric Power, South China University of Technology, Guangzhou Guangdong 510640, China,
E-mail: glm_2008@163.com
[1]Zeng Jun is with the College of Electric Power, South China University of Technology, Guangdong Guangzhou, 510640, China，
E-mail: junzeng@ scut.edu.cn
[1]Wu Jie is with the College of Electric Power, South China University of Technology, Guangdong Guangzhou, 510640, China ,
E-mail: epjiewu@scut.edu.cn
[1]Li Min is with the College of Electric Power, South China University of Technology, Guangdong Guangzhou, 510640, China ,
E-mail: limin_pub@163.com

Abstract –With the development of renewable energy technology, the distributed wind-PV power system has a wider application. This paper proposes a distributed wind-PV power system based on Multi-Agent, whose main character is energy management, and describes the multi-agent cooperative reinforcement learning process using the joint action learning pattern as the cooperative strategy. The experiment of a distributed wind-PV power system shows the efficiency.

Keywords - distributed power; multi-agent; reinforcement learning; Q-learning; joint action learning.

I. INTRODUCTION

Environmental pollution, depletion of fossil fuels has seriously affected the survival of mankind. To change the energy consumption structure and keep energy supply in the path of sustainable development has become a consensus. People around the world are paying attention to renewable energy. Accelerating the development of renewable energy, will be the practical requirements of optimizing energy structure and protecting energy security, and also the urgent needs to protect environment, especially the atmospheric environment. Recently, the distributed power technology based on renewable energy has been rapidly developed. Solar and wind are two kinds of widely used renewable resources. The wind-PV power system based on the complementary characteristics between these two renewable resources has become a hot technology research.

Agent is computing entity or functional unit, which can autonomically perceive information, and generate the corresponding programming through the decision-making and reasoning, and act on the environment. In this paper, each wind and solar system as a separate agent constitutes energy management system (EMS) based on multi-agent. Agent has the competences of learning, coordination, flexibility and autonomy. In EMS, we use reinforcement learning techniques to the research of multi-Agent cooperative learning algorithm.

In recent years, the research on Multi-Agent cooperative reinforcement learning attracts widespread attention. On the one hand, due to the limited capacity of a single agent, it is difficult to complete large-scale complex task. Through collaboration, coordination and consultation, the combination of multiple agents will greatly enhance the intelligence of system. On the other hand, with the gradual popularization and the rapid expansion of internet, agents on network have naturally formed a MAS system. Therefore, the research based on multi-agent learning approach seems particularly urgent. However, in most of the cooperative learning research, only one Agent is learning. For instance, Tan[1] puts forward that using three kinds of cooperative reinforcement learning in cooperative multi-Agent environment. CAI Qingsheng and Zhang Bo put forward a reinforcement learning model based on an agent team. Their common ground is that there is only one agent learning at the same time[2]. In order to realize the cooperative learning, this paper proposes a multi-agent joint action reinforcement learning algorithm. Distributed point of view, each agent should not only consider its own actions, but also the other agent's actions and strategies.

II. REINFORCEMENT LEARNING

Reinforcement learning is a non-supervised learning method which is different from the supervised learning. In the reinforcement learning process, agents could improve their own actions through interacting with the environment, and think of learning as a testing and evaluating process[3]. The basic principles of reinforcement learning technologies are: During learning process, if one action could make the environment to give the system a plus reward, the trend of this action produced by the system will be strengthened, and contrarily it will be weakened. Reinforcement learning can be described as that: Under the environment of discrete-time, finite-state, and finite-actions congregation, it will maximize the cumulative discount reward which is obtained by agents. In this case, the issue of reinforcement learning can use Markov Decision Process (MDP) to model. MDP is defined as a quaternion array (S, A, R, P), where, S for the finite-state set; A for the finite-actions set; R for reward function; R: $S\times A\rightarrow r$, for the mapping from state-action combination to real number; P: $S\times A\rightarrow\Delta$ for transformable function, Δ for probability distribution of state space S.

Q learning is one of the main algorithms of reinforcement learning, and it is a form of model-free reinforcement learning. Q function is defined as the strengthened cumulative discount reward which is obtained through executing action a at the state s, and after this executing the best action sequence. The object of Q learning is to look for a strategy which can maximize the reward in the future. The optimal Q value can be expressed as , defined as the reward summation that obtained through implementing correlative actions and then the optimal strategy, which is defined as follows:

$$Q^*(s,a) = \gamma \sum P(s,a,s') \max Q^*(s',a') + r(s,a) \qquad (1)$$

Where, $P(s,a,s')\sqrt{a^2+b^2}$ for the probability, transformed from state s to s' when executing action a; $r(s,a)$ for the reward obtained by executing action a at state s, γ for discount factor. The update equation of Q function is defined as follows:

$$Q_{(s,a)} = (1-\alpha)Q_{(s,a)} + \alpha[r + \gamma \max_{a' \in A} Q_{(s',a')}] \qquad (2)$$

Where, α ($0 \le \alpha < 1$) for the learning rate; $Q(s',a')$ for the Q function value of Agent executing action a' at state s' [4][5].

III. System Architecture

The distributed wind-PV power system consists of wind turbine, solar cell, and storage battery. Because of its small-scaled system and its disperse space, it is difficult to use concentrative providing energy. This paper takes every power subsystem for an intelligent Agent. Each subsystem consists of perception module, communication module, learning module, knowledge base, decision-making module, executing module, as shown in Fig. 1.

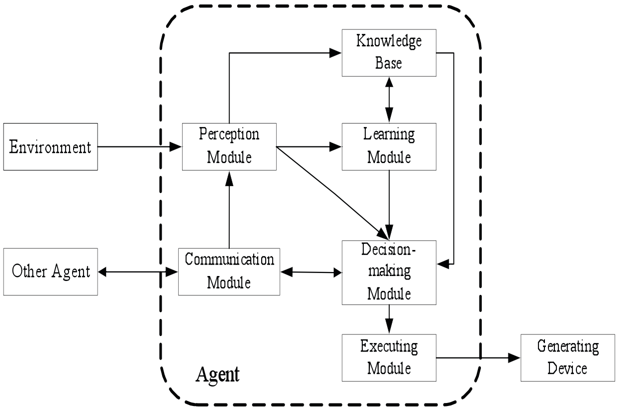


Fig. 1: Block diagram of multi-agent system

IV. Descriptio Of Cooperative Algorithm

In a multi-Agent system, the environment is dynamically changing, and other Agents' behaviors are unknown, so it is almost impossible to build a complete priori model. And many field knowledge is gradually obtained through interacting between Agent and other Agents. Multi-Agent cooperative reinforcement learning means that many Agents reciprocally communicate and cooperate in the process to pursuing a common object. Because the Agents change their own states and environment after obtaining information, every Agent gets the influence from other Agents' knowledge, beliefs, intentions and so on during the learning process.

The distributed wind-PV power system is such a multi-Agent system that is in the dynamically changing environment. In order to overcome its disadvantages, for example, without complete priori model and knowledge, and single agent's uncompleted learning, this paper proposes a Joint Action Learning (JAL) model. In this model, the current action that one Agent is executing is the optimal response to one of other Agents' congregations of actions. Because this paper is discussing a distributed Multi-Agent system, each Agent in this system is indistinctive. Here, the JAL is a learning manner which is based on the forecast that each Agent toward other Agents' actions. According to the system structure proposed before, the learning module is shown in Figure 2.

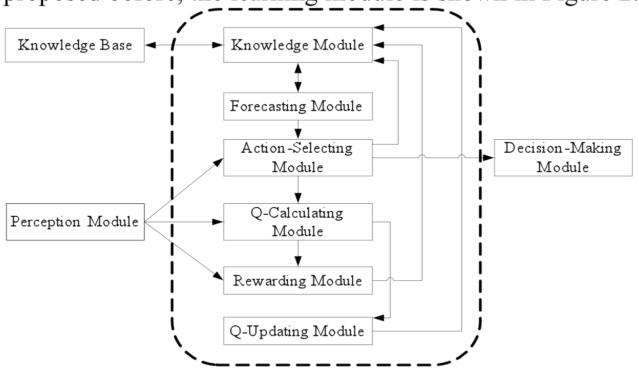


Fig. 2: Block diagram of learning module

The cooperative reinforcement learning algorithm this paper proposed is described as follows:

- Initializing all Agents' Q value in the Q-Updating Module to zero, for Agent i□i=1□2□3□…,n□, its finite-action congregation is A ;
- Agent i obtains the current state $s \in S$, S is Agent's finite environment state congregation;
-
- In the Forecasting Module, according to the current state s, other Agents' action-executing probability
- $$P_i^{a_k} = \frac{C_j^{a_k}}{\sum_{a_m \in A_j} C_j^{a_m}}'$$
- (that is action a_k 's probability of Agent j, $C_j^{a_k}$ is the times of $a_k \in A_j$ in Agent j)stored in Agent i's Knowledge Module, and historical Q value, Agent i will presume other Agents' actions at state s, so that form a forecast action congregation Λ^{-i} ;
-
- In the Action-Selecting Module, Agent i will select the current most optimal action a_i^* , according to the following action selecting strategy:
- $$\Lambda^* = \Lambda^{-i} \cup \left\{ a_i^* \right\} = \arg \max_{a_i \in A_i} Q(s, a_1, \ldots, a_i, \ldots, a_n) \qquad (3)$$
- Executing the action a_i , it will obtain the new state s' and reward r from environment;
- In the Q-Calculating Module, the values obtained above will be substituted into following formula to update Q value, and then the result will be stored in the Q-Updating Module;
- $$Q_{(s,a_1,\ldots,a_i,\ldots,a_n)} = (1-\alpha)Q_{(s,a_1,\ldots,a_i,\ldots,a_n)} + \alpha[r + \gamma \max_{a_i' \in A_i} Q_{(s',a_1,\ldots,a_i',\ldots,a_n)}] \qquad (4)$$
- Each Agent will store its updated data in the Knowledge Module into the Knowledge Base, and then incept the updated information of other Agents' Knowledge Base through communicating;
- One learning process is over, it will wait or enter next learning process at once.

IV. Application Example

In this paper, the distributed wind-PV hybrid power system in the New Energy Research Center of South China University as our research background, we will analyze the cooperative learning process. This system consists of six wind turbines and four photovoltaic cells (PV), with a total capacity of 70KW. The quaternion array in this paper is defined as $S=\{W_{spead},W_{dir},I_{sun},L_{need},S_{equip}\}$, where W_{spead} for wind speed, W_{dir} for wind direction, I_{sun} for sunlight, L_{need} for load requirement, S_{equip} (including four states, that is hot-standby, cold-standby, downtime, and network) for current state of wind turbine or PV. This paper only considers the wind turbines and PV at the hot-standby state, so each Agent's action set is $A=\{a_1,a_2\}$(a_1 for joining in the generation queue, a_2 for not joining in the generation queue). This paper takes one decision-making process for a learning process. Each decision-making may be initiated by the user Agent or other Agent, so the learning process we discussed here is a decision-making process initiated by different Agent asynchronously. Here the Q value will take no account of the impact of the future value. So, the discount factor γ =0, the reward R is decided by three factors together, that are: whether balance between supply and requirement (R_1), power quality (R_2), as well as the electrical price (R_3).

$$R=\omega_1 R_1+\omega_2 R_2+\omega_3 R_3 \qquad (5)$$

Where, $R_1=\begin{cases}10 & Balance\\ -5 & Unbalance\end{cases}$

（ R_1 for the reward of the joint actions）

$$R_2=\begin{cases}10 & High\\ 5 & Medium\\ 0 & Low\end{cases}$$

$R_3=\dfrac{1}{P}$ (P for the electrical price)

We set the learning rate α =0.5, the discount factor γ =0, ω_1 =0.5, ω_2 =0.3, ω_3 =0.2, and initialize all Q values to zero. We suppose at one period of time all Agents' output power are rated capacity, and during this period of time the power quality and the electrical price of each Agent have been given and shown in Table 1.(WT: Wind Turbine, PV: Photovoltaic cells, PQ: Power Quality, EP: Electrical Price)

Table 1: System parameters

Name	Type	Capacity	PQ	EP
Agent1	WT	15	High	0.6
Agent2	WT	10	Medium	0.68
Agent3	WT	7.5	High	0.7
Agent4	WT	5	High	0.8
Agent5	WT	15	Medium	0.65
Agent6	WT	7.5	Low	0.6
Agent7	PV	1	High	3.0
Agent8	PV	2	Medium	2.8
Agent9	PV	3	Medium	2.5
Agent0	PV	4	Low	2.0

Since at first each Agent's Knowledge Base is empty, it needs training for long time to enrich the Knowledge Base. The initial action selecting will not follow the optimal strategy, so it should find the optimal strategy through continuous exploring. In this paper, we need to seek for the optimal strategy through following task decomposition process of a decision-making process, and update the Q value. Here we take a task 50KW initiated by the load Agent for example, and the detailed process as shown in Fig. 3.

In the above task decomposition process, the first column represents the Agents have joined in the task queue, and the second column represents the residual requirement quantity after the frontal Agent joining in the task queue, and when negative appears it will return to front and newly pass to the next Agent to distribute the task, at the same time the Q value will be updated in the third column, until the residual requirement quantity is zero this process will end. In order to achieve the final purpose of optimization, we will learn many times the decision-making process that is at the same state. Each process will use the random exploring method at all times, until it finds a decision-making process that is different from the frontal result, and then store these results in each Agent's Knowledge Base. In what follows we will list partial storage strategy, as shown in Figure 4. After a lot of the learning process, each Agent's Knowledge Base has the stored learning result. In Fig.4, （50，S）represents the load requirement and other current states. After every decision-making process ended, each Agent's Knowledge Base will update the action-executing probability of other Agents. Till the decision-making process in Figure 4 ended, the updated executing probability of action a_1 is shown in Table 2.

Table 2: Rate of action executing

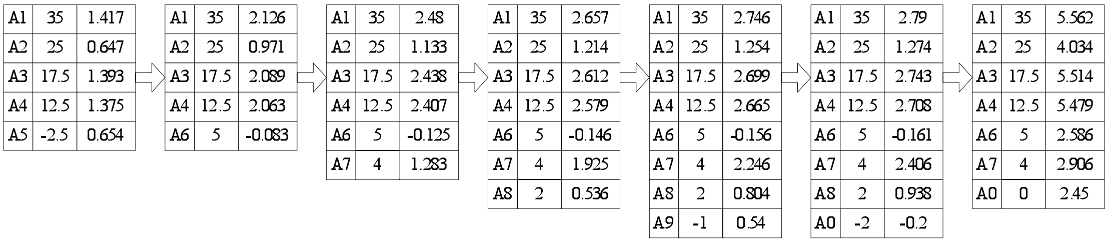
Agent	A1	A2	A3	A4	A5
Pa_1 （%）	77.78	88.89	66.67	66.67	66.67
Agent	A6	A7	A8	A9	A0
Pa_1 （%）	66.67	66.67	55.56	55.56	66.67

After enriching the Knowledge Base for period of time, assuming the load Agent initiates a request of 50KW again, each Agent will do the decision-making according to the cooperative reinforcement learning algorithm. We will take Agent 1 for example, it will firstly select several congregations according to Table 2, and then evaluate which is better and decide whether to join the generation queue according to the factors of whether balance between supply and requirement, and the historical Q value in the Knowledge Base. As the following process, from (6) to (8), Agent 1 finally decides to join in the generation queue. In each Agent, it runs such an algorithm to decide whether to join in this queue, at the same time carries out the Q learning for this decision-making process, and stores the result in the Knowledge Base.

$$\Lambda^{-1}=\{A2,A3,A4,A6,A7,A0\}\Rightarrow \qquad (6)$$
$$\Lambda_1=\{A1,A2,A3,A4,A6,A7,A0\}(Q:4.07586)$$

$$\Lambda^{-1}=\{A2,A4,A5,A7,A0\}\Rightarrow \qquad (7)$$
$$\Lambda_2=\{A1,A2,A4,A5,A7,A0\}(Q:4.41467)$$

$$\Lambda^{-1} = \{A2, A3, A4, A5, A6, A7, A0\} \Rightarrow \qquad (8)$$
$$\Lambda_3 = \{A2, A3, A4, A5, A6, A7, A0\}(Q: 4.11157)$$



A1	35	1.417
A2	25	0.647
A3	17.5	1.393
A4	12.5	1.375
A5	-2.5	0.654

A1	35	2.126
A2	25	0.971
A3	17.5	2.089
A4	12.5	2.063
A6	5	-0.083

A1	35	2.48
A2	25	1.133
A3	17.5	2.438
A4	12.5	2.407
A6	5	-0.125
A7	4	1.283

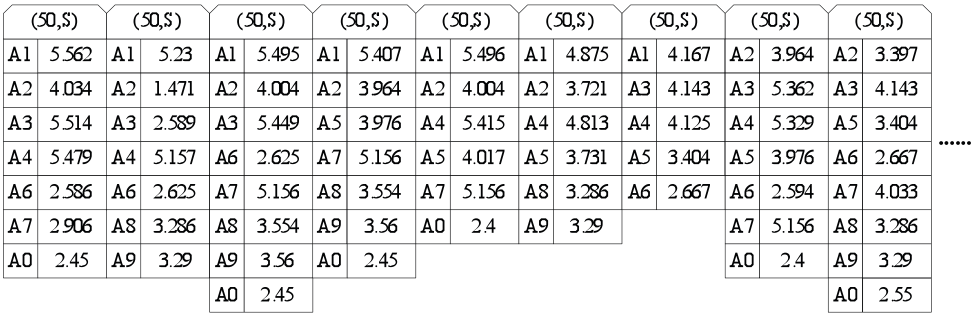
A1	35	2.657
A2	25	1.214
A3	17.5	2.612
A4	12.5	2.579
A6	5	-0.146
A7	4	1.925
A8	2	0.536

A1	35	2.746
A2	25	1.254
A3	17.5	2.699
A4	12.5	2.665
A6	5	-0.156
A7	4	2.246
A8	2	0.804
A9	-1	0.54

A1	35	2.79
A2	25	1.274
A3	17.5	2.743
A4	12.5	2.708
A6	5	-0.161
A7	4	2.406
A8	2	0.938
A0	-2	-0.2

A1	35	5.562
A2	25	4.034
A3	17.5	5.514
A4	12.5	5.479
A6	5	2.586
A7	4	2.906
A0	0	2.45

Fig. 3: Process of task decomposing and Q-value updating



(50,S)		(50,S)		(50,S)		(50,S)		(50,S)		(50,S)		(50,S)		(50,S)		(50,S)	
A1	5.562	A1	5.23	A1	5.495	A1	5.407	A1	5.496	A1	4.875	A1	4.167	A2	3.964	A2	3.397
A2	4.034	A2	1.471	A2	4.004	A2	3.964	A2	4.004	A2	3.721	A3	4.143	A3	5.362	A3	4.143
A3	5.514	A3	2.589	A3	5.449	A5	3.976	A4	5.415	A4	4.813	A4	4.125	A4	5.329	A5	3.404
A4	5.479	A4	5.157	A6	2.625	A7	5.156	A5	4.017	A5	3.731	A5	3.404	A5	3.976	A6	2.667
A6	2.586	A6	2.625	A7	5.156	A8	3.554	A7	5.156	A8	3.286	A6	2.667	A6	2.594	A7	4.033
A7	2.906	A8	3.286	A8	3.554	A9	3.56	A0	2.4	A9	3.29			A7	5.156	A8	3.286
A0	2.45	A9	3.29	A9	3.56	A0	2.45							A0	2.4	A9	3.29
				A0	2.45											A0	2.55

......

Fig. 4: Storage of repository

VI. CONCLUSIO

At Present, the domestic status, that is the power shortage and the continued growth of oil consumption, makes the use of renewable energy very promising, so that the distributed wind-PV power system will be an economical and reasonable power supply pattern. Using the Multi-Agent technology in the system for the distributed energy management system is of great significance. The research of Multi-Agent system's cooperative mechanism usually emphasizes the single learning of Agents and takes no account of other Agents' actions, so that the MAS lacks the cooperative mechanism. This paper proposes a Multi-Agent cooperative reinforcement learning algorithm—the joint action reinforcement learning algorithm. In the algorithm, each Agent forecasts its own action strategy through observing the historical actions of other cooperative ones and makes the corresponding decision-making to achieve the optimal joint action strategy. This paper carries out analyse and research with this algorithm to the distributed wind-PV power system, and shows the feasibility of the algorithm.

ACKNOWLEDGMENT

Project supported by the State Key Program of National Natural Science Foundation of China (No.60534040)

REFERENCES

[1] Tan Ming. Multi-agent reinforcement learning: independent vs cooperative Agents. In :Proceedings of the 10th International Conference on Machine Learning (ICML293) ,1993.330～337

[2] CAI Qingsheng, ZHANG Bo. An agent team based reinforcement learning model and its application. Computer research and development,2000 ,37 (9)

[3] Stuart Russell, Peter Norvig write, JIANG Zhe, JIN Yimin translate. Artificial Intelligence-a modern method (second edition)[M]. Beijing: People post electric publishing company,2004

[4] Tom M Mitchell. Machine L earning [M]. Beijing: China Machine Press: 2003. (in Chinese)

[5] Warkins C , Dayan P. Technic note : Q - Learning [J] . Machine Learning , 1992 , 8 : 279 - 292.

[6] Claus C, Boutilier C. The dynamics of reinforcement learning in cooperative multi-agent systems〔C〕. In: Proceedings of the Fifteenth National Conference on Artificial Intelligence, 1998, 746～752.