Supplementary of Paper "A Stochastic Game Framework for Efficient Energy Management in Microgrid Networks"

Contents

1 Transaction of power among microgrids

In our model, sellers quote a price while buyers are assumed to adhere to the price determined by the sellers. After the microgrids select their respective p_t^i and u_t^i actions, they are divided into two groups namely buyer microgrids and seller microgrids. A grid is classified as a buyer microgrid based on whether or not the value of u_t^i selected is negative. Conversely, a grid is classified as a seller microgrid if the value of u_t^i selected is positive.
Once the microgrids are divided into groups of buyers and sellers, energy trading happens in the following way. First, a microgrid from the seller group (let's call it the leader), that quotes the lowest price is selected. The amount of energy that the leader microgrid is willing to sell is shared amongst the buyer microgrids, proportional to the energy they demand. This is to ensure that there is no bias amongst buyer microgrids. Once the leader microgrid has sold all of its energy, a new leader is chosen, i.e., the one quoting the next best price. This chain continues on till there are no seller microgrids or no buyer microgrids left in the process.

2 ADL Scheduling

ADL (Activities of Daily Living) are tasks that have to be fulfilled within a certain time frame, instead of requiring power immediately. This can be further illustrated by the following example: Imagine an office goer who leaves at 9 a.m in the morning and returns at 5 p.m. in the evening. The office goer wants her clothes to be washed before she comes back. With the advent of smart washing machines, it is possible to schedule a task that has to be completed within the stipulated time frame. The washing machine then requests the microgrid to provide adequate energy to fulfill the task. Depending on other factors the microgrid decides to fulfill this request or defer it to a different time within the stipulated time frame. As time passes, the priority of the request increases and ultimately the washing machine is provided with the required energy. ADL scheduling does not reduce the total energy consumption, it merely shifts the peak load, thereby preventing the microgrid from being overloaded.
In our experiments, we have experimented with cases where the ADL demands are fixed for each day as well as ADL tasks that vary on a day to day basis. These ADL demands are specified at the start of each day to each microgrid. The configuration for the experiments we have run are as follows:

a. Configuration for fixed ADL demand: For the fixed ADL demand setting, 3 ADL tasks have to be satisfied by the microgrid. The amount of electricity required to fulfill the ADL task as well as the time period within which the task must be completed is combined together in the form of a tuple. The configuration for the fixed ADL demand is as follows : [(1,2),(1,3),(2,4)], where the first term in the tuple represents the amount of electricity required and the second term represents the time interval by which the given amount of energy has to be fulfilled.

b. Configuration for the Variable ADL demand: For the variable ADL demand, the microgrids receive 0 to 3 ADL tasks at the start of the day. The number of ADL tasks as well as the time period by which these tasks have to be completed is determined by a probability matrix. The amount of electricity required for each of these ADL tasks can be one of these 3 units: 0, 1 or 2. Similar to the above case, the amount of electricity required to fulfill the ADL task as well as the time period within which the task would have to be completed is combined together in the form of a tuple.

Binary encodings for the ADL tasks: For easy representation of the ADL tasks that have to be fulfilled, a binary representation is used. In our binary representation, 1 implies that that particular task has to be fulfilled and 0 implies that that particular task is not to be considered at that particular time step. An example of this is as follows: Let's assume that the first and third ADL tasks have to be completed. The binary representation of this would be 101 as the microgrid would be interested in fulfilling only the first and 3rd ADL task at that particular time step. This binary representation is fed into the network in a decimal format (which in the example highlighted above would be 5). Moreover, the binary representation also takes care of the time period within which the tasks would have to be fulfilled by ordering the ADL tasks in a sequential manner, i.e, the first digit of the binary tasks corresponds to the fourth time step. Similarly, the second digit corresponds to the second time step and so on.

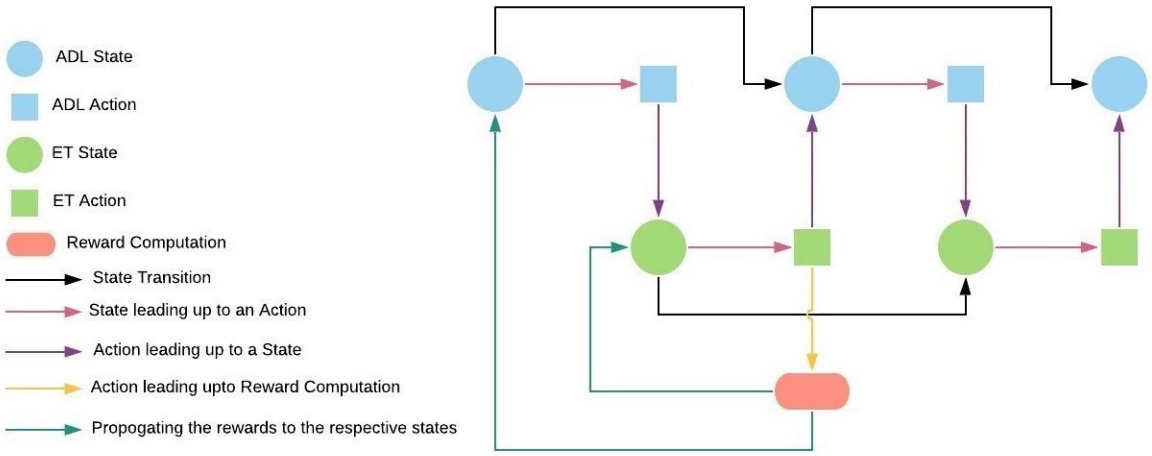
ADL State
ADL Action
ET State
ET Action
Reward Computation
State Transition
State leading up to an Action
Action leading up to a State
Action leading upto Reward Computation
Propogating the rewards to the respective states

Figure 1: Interplay of states and actions between ADL agent and ET agent

The ADL State in the ADL Network is a decimal format representation of the ADL tasks remaining for the day. The ADL network picks an action that is present within the superset of the ADL tasks remaining throughout the day and passes that information to the ET Network in a decimal format. The ET network then uses this information for further decision making.

3 Proposed Algorithm

To fulfill the demand-side management tasks as well as supply-side tasks, each microgrid employs two agents. The first agent (also called the ADL agent) is responsible for the demand-side management. It decides which ADL tasks would be scheduled in the current time step, and this information is then provided to the second agent. The second agent (also called the Energy Trading (ET) agent) is responsible for the supply-side management. It decides the units of electricity to buy or sell, and also sets the transaction prices, i.e., the prices at which the energy trading happens.
Based on the actions taken by the ADL and ET agents, a common reward is obtained by both the agents. This can be justified by the reasoning that both the agents are cohesively working in order to fulfill a common higher goal. Hence, the same credit will be assigned to both of these agents. Due to the interplay between the ADL and ET agents, a single MDP is created which models the state transitions, action selection as well as the reward computation for both the agents. This interplay is shown in Figure 1.

The advantages of using the two separate networks that share the same rewards are as follows: (a). By creating two networks that perform two different tasks that help fulfill a common goal, we have devised a method to successfully model the execution of sequential tasks, using RL. Moreover, by propagating the same reward to both the networks, we have also empirically shown that sharing the same reward for modeling sequential tasks does lead to network learning. Such kind of a sequential learning approach can be used for a lot of real-world fields such as robotics, auctions etc (b). By creating two networks, instead of one very large network, we reduce the number of iterations needed to obtain optimal policy as this enables better exploration of the action space (c). The interplay between the networks as shown in the paper is also novel to the best of our knowledge. Note that both, the ADL

Agent and the ET agent have the same state space except for one parameter. The ADL agent has a parameter known as the ADL state (which signifies which ADL actions have to be fulfilled). Instead of this parameter, the ET agent has a parameter known as ADL action (the action chosen by the ADL agent). Hence the replay buffers for both the agents are similar. Therefore, by sharing a similar state space and reward, the agents are cooperating. Moreover since the ADL and ET agents have to optimize (increase) their rewards, they would implicitly cooperate to obtain an optimal policy.

4 Battery Scheduling and Energy Trading Constraints

For emulating a real world scenario, it becomes imperative that real world constraints are imposed on the amount of energy bought or sold. These constraints are dependant on physical limitations such as the maximum battery capacity, the max energy that can be handled by each microgrid etc. The constraints are imposed as follows:
a. Lower bound on the amount of electricity traded:

$$u_t^i \geq max(-M,\ ne_t^i - F(A_t^i) - B). \tag{1}$$

The first term $-M$ depicts that a microgrid cannot be allowed to buy more than M amount of electricity, thus preventing the microgrid circuits from being excessively overloaded due to the inflow of excess energy. Thus $-M \leq u_t^i$.
After each transaction, the amount of energy that would be stored in the battery of each microgrid, (after factoring in the energy generated, the Non-ADL demand, the ADL demand, the ADL action selected and the energy present in the battery prior to the transaction) would be less than or equal to the the maximum battery capacity, hence preventing the microgrids from buying excess energy and then in turn, wasting it. Thus,

$$ne_t^i - F(A_t^i) - u_t^i \leq B, \text{ which implies} \tag{2}$$

$$ne_t^i - F(A_t^i) - B \leq u_t^i, \tag{3}$$

where $F(A_t^i)$ represents the units of energy that are required to fulfill the selected ADL action.
The second term in the max function in (1) ensures that the ADL action selected by the ADL Network is fulfilled.
A maximum of the above two terms is taken to allow the microgrid to trade the maximum energy possible whilst fulfilling the decided ADL actions and also taking the microgrid stability into consideration as well.
b. Upper bound on the amount of electricity traded:

$$u_t^i \leq ne_t^i + d_t^i - F(A_t^i). \tag{4}$$

The upper bound is derived from the fact that once an ADL action has been chosen then it has to be satisfied by the microgrid. Thus, the amount of energy that the microgrid should possess after trading energy should be greater than or equal to $F(A_t^i)$.
Thus,

$$ne_t^i + d_t^i - u_t^i \geq F(A_t^i), \text{ which implies} \tag{5}$$

$$ne_t^i + d_t^i - F(A_t^i) \geq u_t^i. \tag{6}$$

After the transactions are completed, the excess energy that remains is stored in the battery for future use. The battery state (b_t^i) is updated as follows:

$$b_{t+1}^i = \max(0, ne_t^i - u_t^i - F(A_t^i)). \tag{7}$$

5 Neural Network Model

Here we explain the exact architecture of the ADL network and the Energy trading network:

5.1 ADL Network Architecture

1. Structure:
    - Zeroth Layer: Five input States: ND, D, T, ADL State, Grid Price
    - 1st Layer: 16 Neurons
    - 2nd Layer: 16 Neurons
    - 3rd Layer: Outputs: 8 outputs, since the maximum ADL jobs we consider here is 3 in both fixed ADL and variable ADL case.

2. State Space

   - ND is the net demand i.e it is the cumulative sum of the battery and the energy generated
   - D is the local consumer demand
   - T is the current time step
   - ADL state signifies the number of ADL demands that have to be fulfilled for the day
   - GP is the grid price

3. Action Space

   - The model outputs the ADL loads that have to be fulfilled in the current time step

4. The network minimizes the following loss function:

$$L(\lambda) = ((R + \gamma * \max_{adl}(Q_{adl}(s_{adl+1}, adl_t^i|\lambda))) - Q_{adl}(s_{adl}, adl_t^i|\lambda))^2.$$

   We use Adam optimizer with learning rate 0.0001, β_1=0.9, β_2=0.999 and ϵ=10^{-8} to update network weights. The discount factor γ is kept at 0.9.

5.2 Energy Trading Network

1. Structure:

   - Zeroth Layer: Five input States: ND, D, T, ADL Action, Grid Price
   - 1st Layer: 32 Neurons
   - 2nd Layer: 32 Neurons
   - 3rd Layer: Outputs: The number of Neurons in the output layer is given by the following formula: *(max battery + max energy generated) * 6 + max energy that can be received + 1*

2. State Space

   - ND is the net demand i.e it is the cumulative sum of the battery and the energy generated
   - D is the local consumer demand
   - T is the current time step
   - ADL action signifies the amount of ADL load that the previous model has decided to fulfill in the current time step
   - GP is the grid price

3. Action Space

   - The model outputs the units of electricity to be bought or sold
   - The model also outputs a price between gp - 5 and gp if it is selling, else it outputs a price of 0.

4. The network minimizes the following loss function:

$$L(\theta) = ((R + \gamma * \max_{et}(Q_{et}(s_{et+1}, p_t^i, u_t^i|\theta))) - Q_{et}(s_{et}, p_t^i, u_t^i|\theta))^2.$$

   We use Adam optimizer with learning rate 0.0001, β_1=0.9, β_2=0.999 and ϵ=10^{-8} to update network weights. The discount factor γ is kept at 0.9.