1 Info on Simulation study

The following Simulations are generated with the class fro the mandetory assignement.In this simulationstudy we have changed the unit of measurement to m^2 instead of Km^2 KWH instead of MWH.We devide the gridprice which is said to have the unit $/KWH by 10.

We conducted an evaluation of the provided microgrid system as part of the mandatory assignment. Our approach involved employing deep reinforcement learning techniques within the Gym framework, utilizing a custom environment tailored to the Gym package. Prior to employing deep reinforcement learning through Stable Baselines 3, we created a bespoke environment using Gym. In this segment of the study, we applied the PPO (Proximal Policy Optimization) algorithm from Stable Baselines,with these parameters,

$PPO("MlpPolicy", env, verbose = 1, ent\_coef = 0.01, learning\_rate = 0.01)$.

The microgrid system features a key function named 'transition,' responsible for tracking the energy stored in the battery in kilowatt-hours (KWh). Additionally, this function updates the '*working_space*' variable, a vector of length 3. Each element of this vector can either be 1 or 0, signifying the status of various energy sources. The first element indicates whether solar generation is active, the second for wind generation, and the last for the generator. These values are updated through the '*action_adjust_workingstatus*' variable, which shares the same length. For example, if '*action_adjust_workingstatus*' is [1, 1, 0], it means the microgrid aims to generate energy using both wind and solar sources.

Furthermore, each energy generation type corresponds to specific action vectors. The energy generated can either support the load, store energy, or be sold back to the grid. These actions are represented in the code as variables named '*wind_action*,' '*solar_action*,' and '*generator_action*.' For instance, if '*wind_action*' is [1, 1, 0], the microgrid utilizes generated wind energy to support the load and charge the battery.

Another action determines whether to purchase energy from the grid. We implemented this action deterministically, based on whether the microgrid produces enough energy to meet the household load or not. Additionally, there's an action to discharge the battery. Both of these actions are binary variables. While it was possible to include a binary variable within '*action_purchase*' to determine whether energy bought from the grid could be stored in the battery, we opted for a simpler approach.

The observation space of the microgrid environment encompasses windspeed, solar irradiance, total household load, the working status, and the State of Charge (SOC) of the battery.

Comparing our policy to random one for all simulations using 25,100 and 250 households,only using solar,we see an improvement in the reward.All plots of total accumulated reward show a more positive reward, which is denoted as the negative of the cost, for the learn RL policy.We see that random policy generates less solar for all simulations in addition to being less intelligent about its energy management when it comes to other energy source,how much to sell back and how much to by from the grid.When we start using other energy sources as well,wind is more popular

The new reward in question 4 is performes wors then the reward used in the first part of the assignement.The reward is defined only trough the amount it buys from the grid.Since we defined our reward as the negative of the cost we want the reward to be large as possible.Since the new cost only is defined through using engry biught from the grid,it should have penelized buying from the grid,but the opposite is happening.Even for small purchases from the energygrid,the cost will increase nonelinearly.Since the microgrid cannot support the whole energy load it will have to buy from the grid,and even if it buys less it will still contrubute the cost.One other reason might be that we haven't run the RL algorithm for lon enough.