Switch-China Assignment

Please complete the following activities on your own. Then come to the next session with answers to the questions at the end, and we will discuss them further. If you are not able to complete these steps, please do what you can and come to the second session anyway. In the second session, we will spend some time troubleshooting and working through the steps below and then discuss the answers to the questions and ideas for future work.

1 Model setup

1.1 Open VS Code

> Before you begin the tutorial, please complete the steps described in the "Pre-Tutorial Instructions" document to install Switch on a laptop computer and learn some background information.

In the pre-tutorial setup, you copied data to a folder called "Switch_Tutorial_Small" on your computer. Now we will work with the simple model called "East_China_small" inside this folder. This is a version of the Switch-China open-source model, with extra capabilities added:

- Use of spinning reserves to ensure there is enough power even if loads or renewable production shift unexpectedly
- Carbon limits from 0% to 200% of current levels by 2048
- Ability to produce hydrogen from electricity
- Options to retrofit power plants to burn hydrogen or capture and store carbon
- Hydrogen fuel cell power plant option (new build)

This model has also been sharply downsized from the original model to solve quickly with open solvers.[1] This model has only two study periods (2038 and 2048), two sample days per period, and 5 zones in eastern China.

> Open VS Code, then choose File > Open Folder… and select the East_China_small folder and click OK. Close the "Getting Started" tab if it appears.
>
> Choose View > Command Palette... > Python: Select Interpreter: Python <version> ('base')

1.2 Install plotting packages (if not done already)

In the pre-tutorial instructions, you were given a command to run, as shown below. If that didn't work successfully the first time, it is worth trying again now. If you are able to install these

---

[1] If you are interested, you can find the code used to add these features and do the downsizing at https://github.com/switch-model/switch-china-open-model/tree/ccs_h2_reserves_study. Running "python prep_data.py" will produce the inputs_short_small directory, which is the same as East_China_small. It will also produce larger versions of the model with the features listed above that you can be used for studies (requires a faster solver).

packages, they will produce some helpful graphs each time you run Switch. But if you can't install them, you can complete this tutorial without them.

```
pip install matplotlib plotnine
```

1.3 Examine modules.txt

In VS Code, take a look in inputs/modules.txt. You will see that this model uses some specialized modules to configure it for the China context (china_modules.tech_plans) and to implement features that were not in the public version of Switch at the time it was setup (study_modules.*). Switch will search for these using the standard method for finding python modules. Most of the time that means it will look for them in the directory where you run "switch" from. And indeed, if you look in the Switch_Tutorial_Small, you will find the china_modules and study_modules packages (these are just ordinary folders with an empty __init__.py file inside to tell Python they are packages, and then with various *.py files inside that implement modules within each package).

You may find it interesting to look at the code of china_modules/tech_plans.py. This is a good example of how you can quickly add extra features to a model.

If you are using Switch 2.0.8, you probably don't need the study_modules package. You could switch to using the latest version of Switch's standard modules by making the following updates in modules.txt. This is optional – Switch 2.0.8 will work just fine with the current modules.txt file too.

Original module	**Replace with Switch 2.0.8 equivalent**
study_modules.gen_build_suspend	switch_model.generators.core.build
study_modules.hawaii_save_results	switch_model.hawaii.save_results
study_modules.gen_retrofits_with_retirement	switch_model.generators.extensions.retrofit
study_modules.hydrogen_supply	switch_model.energy_sources.hydrogen.production

1.4 Update options

> Open options.txt and make sure it includes the following two tags (they can be on the same line or separate lines):

```
--solver highs
--solver-options-string "solver=ipm run_crossover=off tech:outlev=1"
```

> Also remove the `--no-save-solution` line because it is obsolete in Switch 2.0.8.

> Save options.txt.

The solver-options string will cause highs to show more output as it runs. You can see the options for highs by running

```
highs -=
```

For many solvers, you can find out what options they accept by running the solver name with "--help", then following up on what it advises. For all ampl-compatible solvers (like the copy of highs that you have), you will be told to run the solver with a "-=" tag to see a list of settings you can use.

One thing to note: every solver accepts different options in the --solver-options-string, and they will generally crash if given the wrong settings. Some people had trouble trying to solve the battery study scenarios using highs in the main tutorial because the options.txt file still had a solver-options-string for cplex. Setting `--solver-options-string "tech:outlev=1"` fixed that, because it was suitable for highs.

2 Activities

2.1 Adjust the model data

The East_China_small model is downscaled from a national model. Unfortunately, we neglected to downscale the limits on nuclear power for these five regions. In addition, the reduced model does not include data on enough renewable energy sites to give a good sense of the potential for these regions. So you will adjust a couple of files before beginning.

> In VS Code, open inputs > total_capacity_limits.csv in the Explorer pane. Change the total_capacity_limit_mw for Uranium to 40055 in both 2038 and 2048.

> In VS Code, open inputs > gen_info.csv. In the second row (Anhui-Central_PV-401), scroll across to the gen_capacity_limit_mw column and change the 8500.0 value to a "." (without the quotes). This will remove the limit for this solar project, which will make it possible to build enough solar for the projects below.
>
> Repeat the step above for the Anhui-Central_PV-401 project in the "gen_info.no_ccs_h2.csv" and "gen_info.no_ccs.csv" files. These will be used for alternative cases below.

2.2 Solve with no carbon limit

The model is currently setup to impose a somewhat arbitrary carbon limit. Let's instead solve it without a carbon limit to see what would be the cheapest system to build and how high the emissions would be.

> Open inputs > carbon_policies.csv (the one with no numbers in the name).
>
> Set `carbon_cap_tco2_per_yr` to a "." for both years. This means no limit is specified. Since there is also no carbon price, switch_model.carbon_policies will now impose no limit on CO_2 emissions, but will still report the total emission level. Your carbon_policies.csv should look like this:

```
PERIOD,carbon_cap_tco2_per_yr,carbon_cost_dollar_per_tco2
2038,.,0.0
2048,.,0.0
```

Be sure to save the file.

Open a terminal pane (Terminal > New) and run the following command to solve the model:

```
switch solve --outputs-dir out_no_carbon_limit
```

(If you get a message that the switch command could not be found, choose View > Command Palette... > Python: Select Interpreter: Python <version> (‘base’), then open a new terminal pane and try again.)

> The model is currently setup to have options to use CCS and hydrogen retrofits and fuel cells. Verify that it chose not to install any of these. This is one way to do that:
>
> - Expand out_no_carbon_limit, then right click on gen_cap.csv and choose “Preview”
> - Filter the Gen_energy_source column to only show “Hydrogen”, then press Enter.
> - Choose the filter option on the Gen Capacity column and make sure the only value shown is 0. (No capacity build with Hydrogen as the energy source.)
> - Click on Gen_energy_source filter, choose “Select all”, then press Enter. This should remove the filter.
> - Click on the filter for Gen_tech, then search for “ccs” and press Enter. This should only show generation projects with “CCS” in the technology name. Make sure Gen Capacity is 0 for all of these.

> Open out_no_carbon_limit > emissions.csv. Make a note of the total emissions (AnnualEmissions_tCO2_per_yr) in 2038 and 2048. (This should be around 1.53e9 in 2038 and 1.23e9 in 2048.)
>
> Open out_no_carbon_limit > electricity_cost.csv. Make a note of EnergyCostReal_per_MWh for 2038 and 2048. (This should be around $71.86/MWh in 2038 and $70.06/MWh in 2048.)

2.3 Solve with basic technologies and a zero-carbon limit

The “inputs/carbon_policies_000.csv” file contains a carbon limit that goes to zero by 2048, but it has the wrong limit for 2038, since it was set for the national model, but you are now running a five-province model.

> Open inputs > carbon_policies_000.csv and set the 2038 CO2 target to 768000000. This should be about half of the 2038 level you found from the previous solution. Also double-check that the limit for 2048 is set to 0.

This file now defines an emission plan that is 50% below least-cost levels by 2038, then zero by 2048.

If you look in the inputs directory, you will also see that there are already alternative versions of gen_info.csv and gen_build_costs.csv that exclude the CCS and H2 technologies. We will use these to see how difficult it would be to reach zero emissions using currently established technologies (i.e., no CCS or hydrogen).

When running multiple scenarios with similar data, it is often useful to use the `--input-alias(es)`, `--exclude-module(s)` or `--exclude-module(s)` options to tell the model to use different input files or modules.

Run the following command to solve the model using your new carbon_policies_000.csv file, omitting the retrofit option (for CCS and hydrogen), and excluding the CCS and H2 generators from your input files (this whole command should be run on one line):

```
switch solve --outputs-dir out_zero_basic --input-alias
      carbon_policies.csv=carbon_policies_000.csv --exclude-modules
      study_modules.gen_retrofits_with_retirement study_modules.hydrogen_supply --input-aliases
      gen_info.csv=gen_info.no_ccs_h2.csv
      gen_build_costs.csv=gen_build_costs.no_ccs_h2.csv
```

> Open out_zero_basic > emissions.csv and verify that the carbon limit was met.
>
> Open out_zero_basic > electricity_cost.csv. Make a note of the total emissions and cost per MWh in 2038 and 2048. (These should be about $83.38/MWh and $103.29/MWh.)

2.4 Solve with hydrogen option and a zero-carbon limit

The previous case had a very high cost for a zero-emissions system. Let's see if using hydrogen could help reduce this cost. You can do this by using alternative versions of input files that are already in the inputs directory. The "no_ccs" versions of gen_info.csv and gen_build_costs.csv include definitions of fuel-cell power plants that can convert hydrogen and electricity directly to electricity and water without combustion, and also for retrofitted versions of coal plants that can burn hydrogen. The gen_retrofits.no_ccs.csv file also has rows indicating which plants the retrofits can be applied to.[2] (These files have "no_ccs" in their name because they contain all the technologies available in the original study except CCS.)

Run the following command to solve a zero-emission scenario with hydrogen options:

[2] When defining retrofits, it's very important to include them in the gen_retrofits file. If they are only listed in gen_info.csv and gen_build_costs.csv, Switch will think the retrofits can be built as cheap, standalone plants, which is usually incorrect. There is more documentation in the retrofits module and at https://github.com/switch-model/switch/blob/master/CHANGELOG.md

```
switch solve --outputs-dir out_zero_hydrogen --input-aliases
      carbon_policies.csv=carbon_policies_000.csv
      gen_info.csv=gen_info.no_ccs.csv
      gen_build_costs.csv=gen_build_costs.no_ccs.csv
      gen_retrofits.csv=gen_retrofits.no_ccs.csv
```

Note that this version of the solve command does not exclude any modules from modules.txt, since we now want to allow retrofits and hydrogen production.

> Open out_zero_hydrogen > electricity_cost.csv. Make a note of the total emissions and cost per MWh in 2038 and 2048. (These should be about $83.37/MWh and $100.98/MWh.)

2.5 Solve a zero-carbon case with hydrogen and demand response (optional)

Now let's see if adding demand-side flexibility could reduce costs further. You can skip this if you are running short of time.

- Open the "switch" source code folder in VS Code or switch to that window if it is open already.
- Navigate to switch_model > balancing > demand_response > simple.py
- Read the documentation at the top of the define_components() to see how to define a demand response option. You will see that there are parameters called dr_shift_down_limit and dr_shift_up_limit that define how much load can be shifted each hour.
- Read the documentation in the load_inputs() function for in this file. You will see that you need to create a file called dr_data.csv with columns for the LOAD_ZONE and TIMEPOINT when loads could be altered and columns to hold the dr_shift_down_limit and dr_shift_up_limit values.

Now let's create the dr_data.csv file needed for this module. Here's one way to do it. (You can use R or pandas instead if you prefer.)

- Open the loads.csv file from the inputs directory in a spreadsheet program like Excel.
- Create new columns called dr_shift_down_limit and dr_shift_up_limit
- Set dr_shift_down_limit to 10% of the zone_demand_mw in the same zone and timepoint.
- Set dr_shift_up_limit to 30% of the zone_demand_mw in the same zone and timepoint.
- Copy the dr_shift_down_limit and dr_shift_up_limit columns and then paste back as values.
- Delete the zone_demand_mw column.
- Save the file as dr_data.csv in the inputs folder. Make sure it is in csv format, not .xlsx.

Now you switch back to the East_China_small window and re-run Switch, adding the demand response option, like this:

```
switch solve --outputs-dir out_zero_hydrogen_dr --include-module
      switch_model.balancing.demand_response.simple --input-aliases
      carbon_policies.csv=carbon_policies_000.csv
      gen_info.csv=gen_info.no_ccs.csv
      gen_build_costs.csv=gen_build_costs.no_ccs.csv
      gen_retrofits.csv=gen_retrofits.no_ccs.csv
```

> Open out_zero_hydrogen_dr > electricity_cost.csv. Make a note of the total emissions and cost per MWh in 2038 and 2048. (These should be about $78.04/MWh and $96.44/MWh.)

3 Questions

This is a highly simplified model, so the findings are not strong enough to use for decisionmaking. However, they do reflect some of the real issues that arise in planning deeply decarbonized power systems.

Please find or think about answer to the following questions. You can use the information given at the end of each section above if you weren't able to run the model yourself.

- How much did the CO2 emissions drop between 2038 and 2048 in the no-carbon-limit case? Why do you think this happened, even though electricity demand went up?
    - You may find it useful to look at dispatch_annual_summary_fuel.pdf in the out_no_carbon_limit directory if Switch produced it.
- What percentage did the electricity costs in 2048 change when a zero-carbon requirement was applied, using basic technology options? Why do you think they went up so much?
- What percentage did 2048 electricity costs change when hydrogen was added as an option? Why do you think costs went down?
- What percentage did 2048 electricity costs change when demand response was added as an option? Why do you think this happened?
- Are there other technologies or policies you would like to evaluate with this kind of model?
    - We can discuss how you might do this during the next session.