Student Project Activity #4: Demand Flexibility Measures

Overview

In this activity, students will propose and evaluate demand flexibility measures for the buildings in the REopt scenario generated from the coincident and diverse projects. Students will also reevaluate their choices of energy conservation measures from Activity #2 based on their new measure choices.

Tasks

1. (20 mins) Choose Measures for demand flexibility. The current list of measures that are packaged with the coincident and diverse projects are the following:

- Add Pre-Cooling or Pre-Heating (measure is named ShiftScheduleByType)
- Add Heat Pump Water Heater
- Add Packaged Ice Storage System
- Add EMS to Control EV Charging

Demand Flexibility Measure Details

On the link in the callout above, click into each directory (which is the measure) and read through the details on what the measures do. Browse the measures and select one that you believe will help you achieve your energy and demand flexibility objective.

2. (30 mins) Add Measure(s). The instructions below are the same as Activity #2.

Details on how to add measures

- Locate the directory in Finder/Explorer: <working>/[coincident|diverse]/mappers/

- Copy the ClassProject.rb file to a new file called ClassProjectDemandFlexibility.rb. Open the file in an editor and replace the `ClassProject < BaselineMapper` line with the new name of the file `ClassProjectDemandFlexibility < BaselineMapper`.

- In the opened ClassProjectDemandFlexibility.rb file find a measure that you would like "enabled" and change the "SKIP" value of your chosen measures from "true" to "false" (i.e., do not skip the measure), shown below:

```
OpenStudio::Extension.set_measure_argument(osw, 'ExampleMeasure',
'__SKIP__', false)
```

- Most measures require arguments to be passed (e.g., to reduce internal lighting loads, the user must pass the percent to reduce the loads by). To edit a measure argument, change the argument as shown below:

```
OpenStudio::Extension.set_measure_argument(osw, 'ExampleMeasure',
'MeasureArgument', <set argument value here>)
```

- Save and close this mapper file

3. (30 mins) Create a new scenario CSV file with REopt functionality.

- Base the project off the previous scenario file (e.g., custom_example_scenario.csv).

```
$ cd <working>
$ uo create -f coincident/class_project_coincident.json -r
coincident/custom_example_scenario.csv

>>> Creating ScenarioFile with REopt functionality, extending from
coincident/custom_example_scenario.csv...

>>> Done
```

- Open the new CSV file which was created (e.g., `REopt_custom_example_scenario.csv`) and substitute all the string inputs under the header “Mapper Class” to from ` URBANopt::Scenario::BaselineMapper ` to `URBANopt::Scenario::ClassProjectDemandFlexibility`

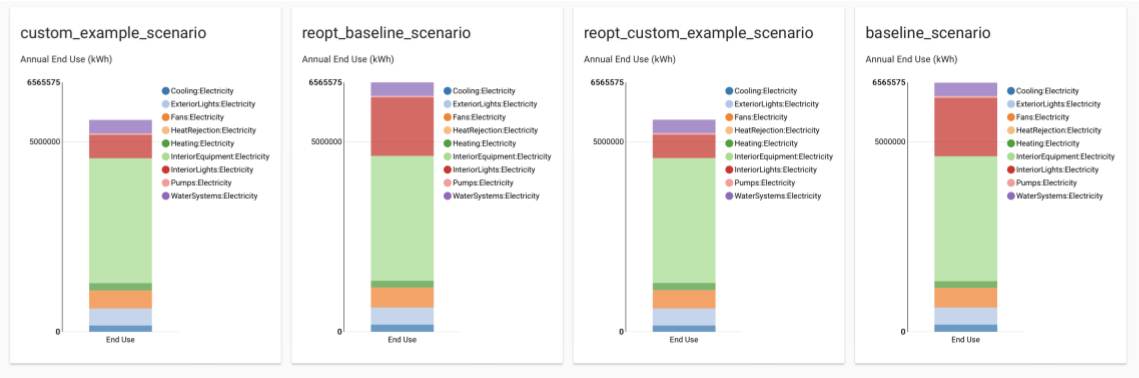
4. (60 mins) Run Command: Run this project model with your new measures added by using the path for your classmeasures_scenario.csv file:

```
$ cd <working>
$ uo run -f coincident/class_project_coincident.json -s
coincident/REopt_custom_example_scenario.csv

>>> Simulating features of 'class_project_coincident.json' as
directed by 'REopt_custom_example_scenario.csv'...

>>> Lots of simulation output
>>> DONE, result = true
>>> Done Running Scenario
```

5. (5 mins) Process Command: Post-process general results for each project scenario.

```
$ cd <working>
$ uo process -d -f coincident/class_project_coincident.json -s
coincident/REopt_custom_example_scenario.csv

>>> Done

# Also run the visualize command to compare baseline with
class_project_coincident.json

$ uo visualize -f coincident/class_project_coincident.json

>>> Done
```

You can view the summary in the `scenario_comparison.html` file that is created in the run directory.



6. (60 mins) Create and process with REopt Commands:

- Process Scenario with REopt Command: Post-process general results for each project scenario.

```
$ cd <working>
$ uo process -r -f coincident/class_project_coincident.json -s
coincident/REopt_custom_example_scenario.csv

>>> Running the REopt Scenario post-processor with scenario
assumptions file:
<working>/coincident/reopt/multiPV_assumptions.json

>>> Post-processing entire scenario with REopt
```

- Process Features with REopt Command: Post-process general results for each project feature (building).

```
$ cd <working>
$ uo process -e -f coincident/class_project_coincident.json -s
coincident/REopt_custom_example_scenario.csv

>>> Running the REopt Scenario post-processor with scenario
assumptions file:
<working>/coincident/reopt/multiPV_assumptions.json

>>> Post-processing each building individually with REopt
```

7. (120 mins) Repeat this process for the other project (coincident and diverse).

8. (60 mins) Assess your Measures: Analyze the effectiveness of your added measures for the coincident and diverse projects. As with the previous activities, the results are in the Project “run” folder.

9. (120 mins) Modify and Iterate: If your chosen measures were not very effective in achieving your energy and demand flexibility goals, choose other measures (or modify existing measure arguments) that you think may have a better chance of success.

- Present a comparison of the cumulative annual electricity consumption for the whole year and plot timeseries hourly electricity consumption profiles over a chosen 3-day period between the baseline scenario and class measures scenario results to display the benefits of your added measures.

10. (120 mins) Write a Final Report: Compile a concise, professional report in which you present your work and findings. Make final recommendations on which energy efficiency or demand flexibility measures to implement for the district based on your modeling and optimization results. Your analysis must include energy and cost comparisons, regardless of your specific evaluation criteria. Use of an executive summary is recommended.

Deliverables (Week 7-8)

- Final ClassProject.rb mapper files used in this project
- Graphical representation of electricity consumption as stated above
- A final report as stated in Task #9