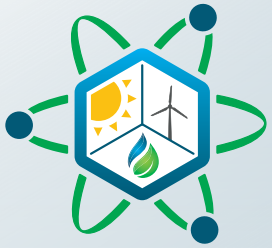


IES
Integrated Energy Systems

LWRS
LIGHT WATER REACTOR SUSTAINABILITY

HERON Workshop Examples

A case-by-case training in running HERON

Dylan McDowell – HERON Developer
Paul Talbot – HERON Owner

INL/MIS-22-65661

Learning by Example

- HERON cases from basic to complex
- Exercises, follow-alongs
- Examples found in `HERON/tests/workshop/htse`

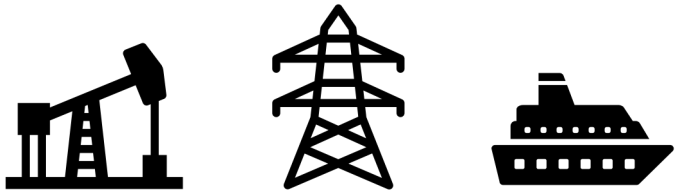
Example

As Simple As They Come

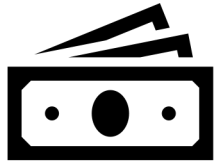
IES
Integrated Energy Systems

Starter Case

- Components
  - NGCC (natural gas electricity generator)
  - Import (imports electricity from external)
  - Grid (demand to be met)

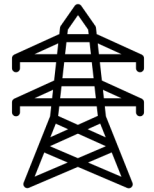


- Objective
  - How big should the NGCC be built to minimize costs?

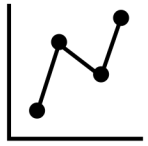


Starter Case

- Guiding Physics/Economics (Drivers)
  - Grid
    - fixed demand to be met
  - NGCC
    - Capital cost for sizing
    - Variable cost for dispatching
  - Import
    - Variable cost (expensive!) for providing electricity



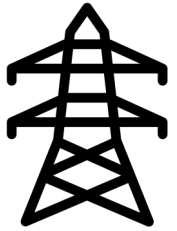
Translating to HERON



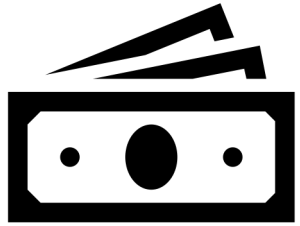
- See `HERON/tests/workshop/htse/example1_simple/heron_input.xml`
- **`<Case>`**
  - Global information about the problem
  - Time shape, discount rates, solvers, etc.
- **`<Components>`**
  - Technical and economic properties of each component
  - Produces and Demands
  - Dispatch: Independent and Fixed
- **`<DataGenerators>`**
  - Synthetic History Data Source

Component by Component

- Grid
  - Technical Specs
    - Demands electricity
    - Fixed dispatch (no flexibility, demand must be met)
    - “Capacity” (demand) depends on synthetic time series
      - Note: time series generators trained separately

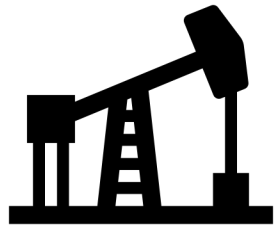


  - Economics
    - None
      - Regulated market example
      - Minimize cost to meet demand

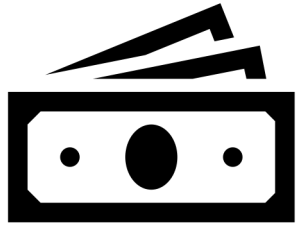


Component by Component

- NGCC
  - Technical Specs
    - Produces electricity
    - Independent dispatch (0 to Capacity each hour)
    - Capacity (electricity) optimized from 10 GW to 40 GW
      - Note: units (GW) are up to the user but should be consistent!

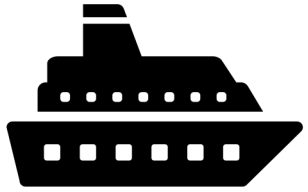


  - Economics
    - Capex, driven by capacity
    - Variable O&M, driven by activity
      - Cost of natural gas, etc

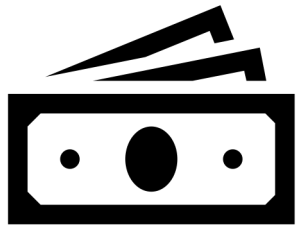


Component by Component

- Import
  - Technical Specs
    - Produces electricity
    - Independent dispatch (0 to Capacity each time step)
    - Capacity (electricity) fixed at large number
      - Note: consider demand to know what fixed cap is enough!

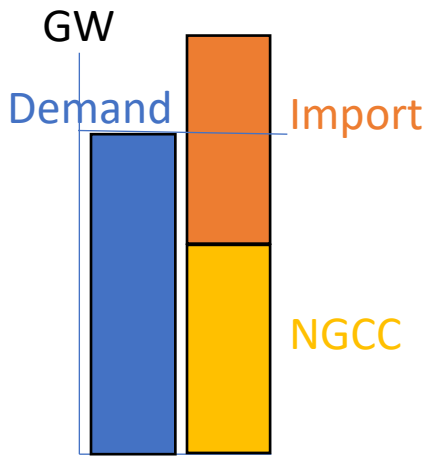


  - Economics
    - Variable O&M, driven by activity
      - Cost of importing electricity



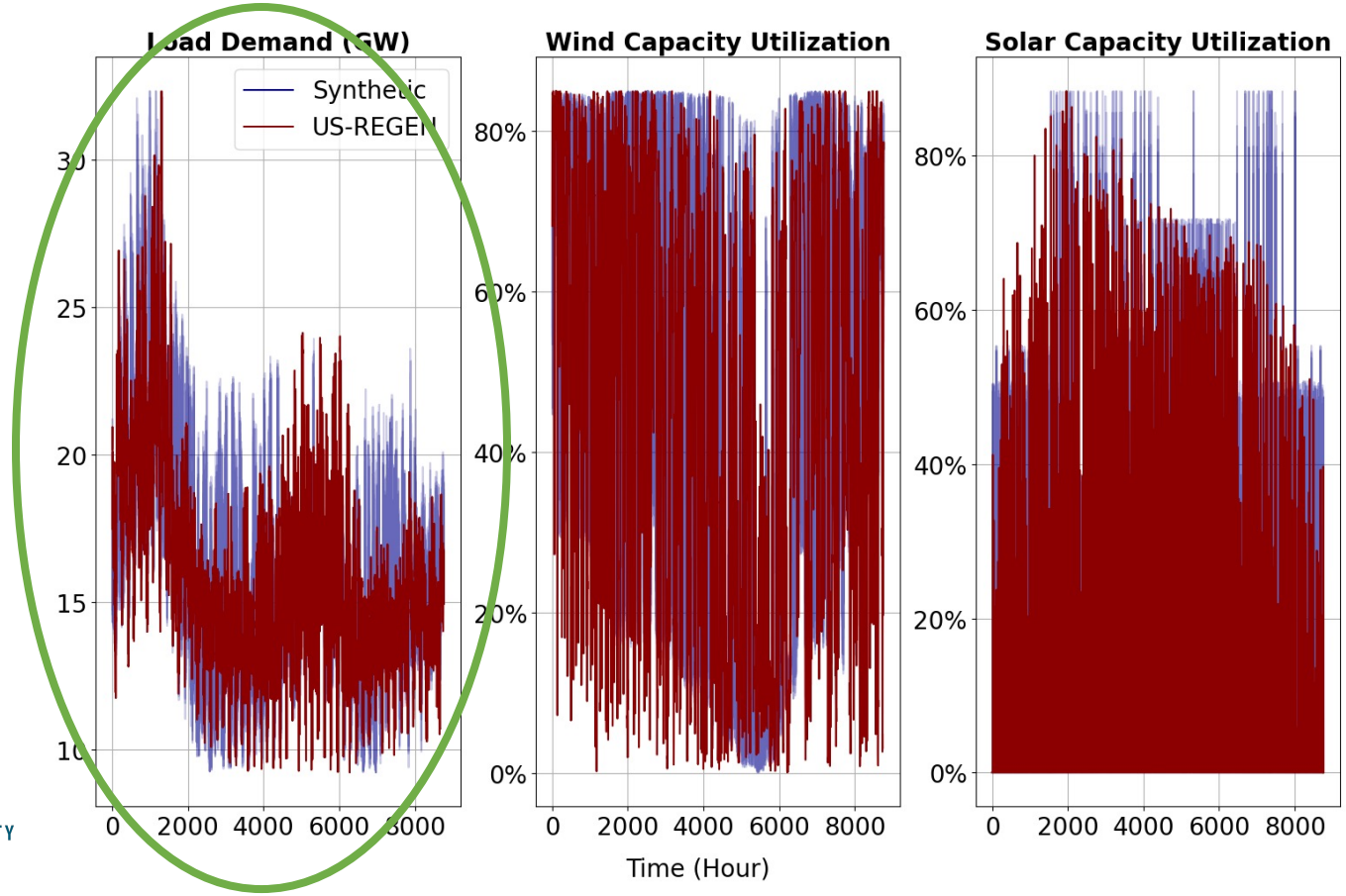
Consider the Case

- Why Import?
  - Why not just have NGCC?
  - If NGCC not enough to meet demand, solve “fails”
    - This doesn’t provide useful feedback to optimizer; it doesn’t know what’s wrong
    - Instead, provide large incentive to meet demand with NGCC
    - Example:



Demand

- Stochastically Trained
- Based on NYISO projections from EPRI



See INL/EXT-21-65473, "A Technical and Economic Assessment of LWR Flexible Operation for Generation/Demand Balancing to Optimize Plant Revenue", Dec 2021

LWRS LIGHT WATER REACTOR SUSTAINABILITY

IES Integrated Energy Systems

Running the Case

- Check Parallelization
  - Available threads: cpus * cores/cpu * threads/core
    - search: how many threads does my computer have [windows/mac/linux]
    - useful parallelization -> limited by inner, outer characteristics
- `<path/to/HERON>/heron heron_input.xml`
- `<path/to/RAVEN>/raven_framework outer.xml`
- It's optimizing, so it will take a bit!

- Things to look at:
  - `1_simple_o/opt_soln_0.csv` (spreadsheet) optimizer progress
  - Screen Output

Things to Look At While Running

- `1_simple_o/opt_soln_0.csv` – progress of the optimizer

`1_simple_o` = case name; `o` = outer; `opt_soln_0.csv` = optimization record

Optimizer Steps (sets of samples)

Component Sizes (opt variables)

Mean NPV (objective)

Status of each step
- Accepted: improvement!
- Rejected: went too far
- Rerun: re-check gradient

	A	B	C	D	E
1	iteration	accepted	ngcc_capacity	import_capa	mean_NPV
2	0	first	11.5	100	-1.423E+10
3	1	accepted	17.5	100	-8.773E+09
4	2	accepted	23.5	100	-7.436E+09
5	3	rejected	11.5	100	-1.423E+10
6	4	rerun	23.5	100	-7.432E+09
7	5	accepted	31.5	100	-7.229E+09
8	6	accepted	36.833333	100	-7.207E+09
9	7	rejected	26.166667	100	-7.292E+09
10	8	rerun	36.833333	100	-7.202E+09
11	9	rejected	29.722222	100	-7.234E+09
12	10	rerun	36.833333	100	-7.204E+09
13	11	rejected	33.092593	100	-7.224E+09

Note our losses are getting smaller!

LWRS LIGHT WATER REACTOR SUSTAINABILITY

IES Integrated Energy Systems

Things to Look At While Running

- Screen Output: How To Read the Matrix
  - Accepted Sample
    - This means a new optimal point has been found!

```
( 7704.86 sec) PointSet                : DEBUG          -> Wrote master cluster file to "opt_soln.csv"
( 7704.87 sec) PointSet                : DEBUG          -> Wrote sub-cluster file to "opt_soln_0.csv"
( 7704.88 sec) PointSet                : DEBUG          -> Printing metadata XML: "opt_soln.xml"
( 7704.88 sec) STEP MULTIRUN           : DEBUG          -> Just collected job   261    and sent to output "opt_soln"
( 7704.88 sec) GradientDescent         : DEBUG          -> ********************************************************************************
( 7704.88 sec) GradientDescent         : DEBUG          -> Trajectory 0 iteration 171 resolving new opt point ...
( 7704.88 sec) GradientDescent         : DEBUG          ->   ... change: -3.520e+05 new:  7.202887e+09 old:  7.203239e+09
( 7704.88 sec) GradientDescent         : DEBUG          ->   .. accepted!
( 7704.88 sec) GradientDescent         : DEBUG          -> Convergence Check for Trajectory 0:
( 7704.88 sec) GradientDescent         : DEBUG          ->   ...  gradient  : False, 3.33e-02 / 1.00e-04
( 7704.88 sec) GradientDescent         : DEBUG          ->   ... objective  : False, 2.17e-05 / 1.00e-08
( 7704.88 sec) GradientDescent         : DEBUG          ->   ... same point : False, 0.00e+00 / 1.00e+00
( 7704.88 sec) GradientDescent         : DEBUG          -> Resetting convergence for trajectory 0.
( 7704.88 sec) GradientDescent         : DEBUG          -> ********************************************************************************
( 7704.88 sec) STEP MULTIRUN           : DEBUG          -> Testing if the sampler is ready to generate a new input
( 7704.89 sec) GradientDescent         : DEBUG          ->   ... Sample point 262: {'ngee_capacity': 39.989015501936606, 'denoises': 20, 'import_co
 100.0}
( 7704.89 sec) CODE MODEL              : Message        -> job "262" submitted!
```

Things to Look At While Running

- Screen Output: How To Read the Matrix
  - Rejected Sample
    - This means the new proposed opt point is worse than the old opt point
    - Reasons: inaccurate gradient, too large step, no better points nearby
    - Happens frequently, especially near the end

```
( 7338.73 sec) STEP MULTIRUN          : DEBUG          -> Just collected job   249    and sent to output "opt_eval"
( 7338.74 sec) PointSet               : DEBUG          -> Wrote master cluster file to "opt_soln.csv"
( 7338.75 sec) PointSet               : DEBUG          -> Wrote sub-cluster file to "opt_soln_0.csv"
( 7338.75 sec) PointSet               : DEBUG          -> Printing metadata XML: "opt_soln.xml"
( 7338.75 sec) STEP MULTIRUN          : DEBUG          -> Just collected job   249    and sent to output "opt_soln"
( 7338.76 sec) GradientDescent        : DEBUG          -> ********************************************************************************
( 7338.76 sec) GradientDescent        : DEBUG          -> Trajectory 0 iteration 163 resolving new opt point ...
( 7338.76 sec) GradientDescent        : DEBUG          ->   ... change:  1.706e+06 new:  7.204945e+09 old:  7.203239e+09
( 7338.76 sec) GradientDescent        : DEBUG          ->   ... rejected!
( 7338.76 sec) GradientDescent        : DEBUG          -> ********************************************************************************
( 7338.76 sec) GradientDescent        : DEBUG          -> Canceling grad jobs for traj "0" iteration "163".
( 7338.76 sec) GradientDescent        : DEBUG          -> * Submitting new opt and grad points *
( 7338.76 sec) GradientDescent        : DEBUG          -> Adding run to queue: {'ngcc_capacity': 39.989015501936677, 'import_capacity': 100.0
```

Things to Look At While Running

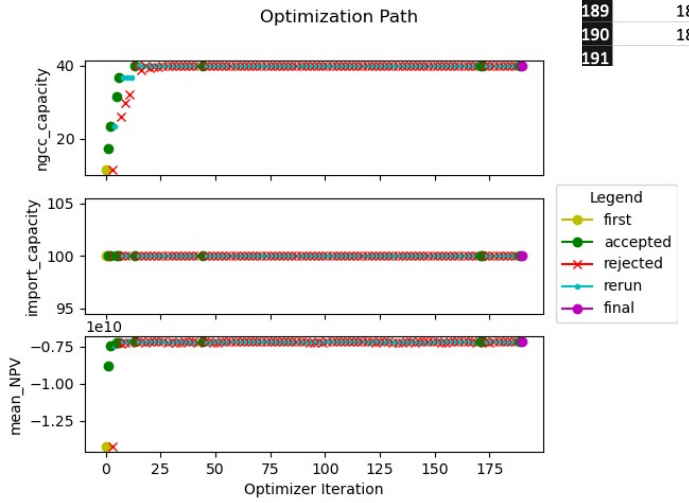
- Screen Output: How To Read the Matrix
  - Rerun Sample
    - Return to best-so-far opt point and rerun the gradient, cut step size
    - Helps to resolve two reasons for rejecting proposed opt points

```
( 7370.07 sec) PointSet                : DEBUG          -> Wrote master cluster file to "opt_soln.csv"
( 7370.09 sec) PointSet                : DEBUG          -> Wrote sub-cluster file to "opt_soln_0.csv"
( 7370.09 sec) PointSet                : DEBUG          -> Printing metadata XML: "opt_soln.xml"
( 7370.09 sec) STEP MULTIRUN           : DEBUG          -> Just collected job   250    and sent to output "opt_soln"
( 7370.10 sec) GradientDescent         : DEBUG          -> ********************************************************************************
( 7370.10 sec) GradientDescent         : DEBUG          -> Trajectory 0 iteration 164 resolving new opt point ...
( 7370.10 sec) GradientDescent         : DEBUG          ->   ... change:  1.956e+06 new:  7.205195e+09 old:  7.203239e+09
( 7370.10 sec) GradientDescent         : DEBUG          ->   ... rerun!
( 7370.10 sec) GradientDescent         : DEBUG          -> ********************************************************************************
( 7370.10 sec) STEP MULTIRUN           : DEBUG          -> Testing if the sampler is ready to generate a new input
( 7370.10 sec) GradientDescent         : DEBUG          ->   ... Sample point 251: {'ngcc_capacity': 39.989015501936677, 'denoises': 20, 'import_
 100.0}
( 7370.10 sec) CODE MODEL              : Message        -> job "251" submitted!
```

Now that it's done …

- Things to look at (see next slides)
- opt_soln_0.csv
  - filter opt CSV by "accepted" to see acceptance path
  - final solution is the last "accepted" point in the CSV
- opt_path.png
  - visualization of opt_soln_0.csv
- screen output

	A	B	C	D	E
171	169	rejected	39.989016	100	-7.21E+09
172	170	rerun	39.989016	100	-7.2E+09
173	171	accepted	39.989016	100	-7.2E+09
174	172	accepted	39.989016	100	-7.2E+09
175	173	rejected	39.989016	100	-7.2E+09
176	174	rerun	39.989016	100	-7.2E+09
177	175	rejected	39.989016	100	-7.21E+09
178	176	rerun	39.989016	100	-7.2E+09
179	177	rejected	39.989016	100	-7.21E+09
180	178	rerun	39.989016	100	-7.21E+09
181	179	rejected	39.989016	100	-7.2E+09
182	180	rerun	39.989016	100	-7.2E+09
183	181	rejected	39.989016	100	-7.2E+09
184	182	rerun	39.989016	100	-7.21E+09
185	183	rejected	39.989016	100	-7.21E+09
186	184	rerun	39.989016	100	-7.21E+09
187	185	rejected	39.989016	100	-7.21E+09
188	186	rerun	39.989016	100	-7.21E+09
189	187	rejected	39.989016	100	-7.2E+09
190	188	rerun	39.989016	100	-7.21E+09
191					




```
-> ****************************************
-> Optimizer Final Results:
->
->  - Trajectory Results:
->   TRAJ    STATUS     VALUE
->     0   converged   -7.204e+09
->     0    active     -7.204e+09
->
->  - Final Optimal Point:
->          mean_NPV        -7.204e+09
->            trajID          0
->       ngcc_capacity       3.999e+01
-> ****************************************
```

LWRS LIGHT WATER REACTOR SUSTAINABILITY

IES Integrated Energy Systems

Things to look at now that it's done

`Screen Output`

- Summary of optimizer status
  - Trajectory that found best point
  - Objective value
  - Opt Vars values

```
-> ****************************************
-> Optimizer Final Results:
->
->  - Trajectory Results:
->   TRAJ    STATUS     VALUE
->     0   converged   -7.204e+09
->     0    active     -7.204e+09
->
->  - Final Optimal Point:
->          mean_NPV          -7.204e+09
->           trajID             0
->       ngcc_capacity         3.999e+01
-> ****************************************
```

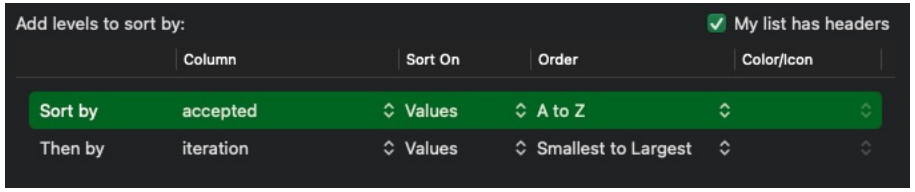
objective

opt var

Things to look at now that it's done

`1_simple_o/opt_soln_0.csv`

- Last "accepted" point is solution
  - 200 iterations is low (good!)
  - Sort by "accepted" then "iteration"



	A	B	C	D	E
1	iteration	accepted	ngcc_capacit	import_capa	mean_NPV
2	1	accepted	17.5	100	-8.77E+09
3	2	accepted	23.5	100	-7.44E+09
4	5	accepted	31.5	100	-7.23E+09
5	6	accepted	36.833333	100	-7.21E+09
6	13	accepted	39.993827	100	-7.2E+09
7	44	accepted	39.989016	100	-7.2E+09
8	171	accepted	39.989016	100	-7.2E+09
9	172	accepted	39.989016	100	-7.2E+09
10	0	first	11.5	100	-1.42E+10
11	3	rejected	11.5	100	-1.42E+10

opt var

objective

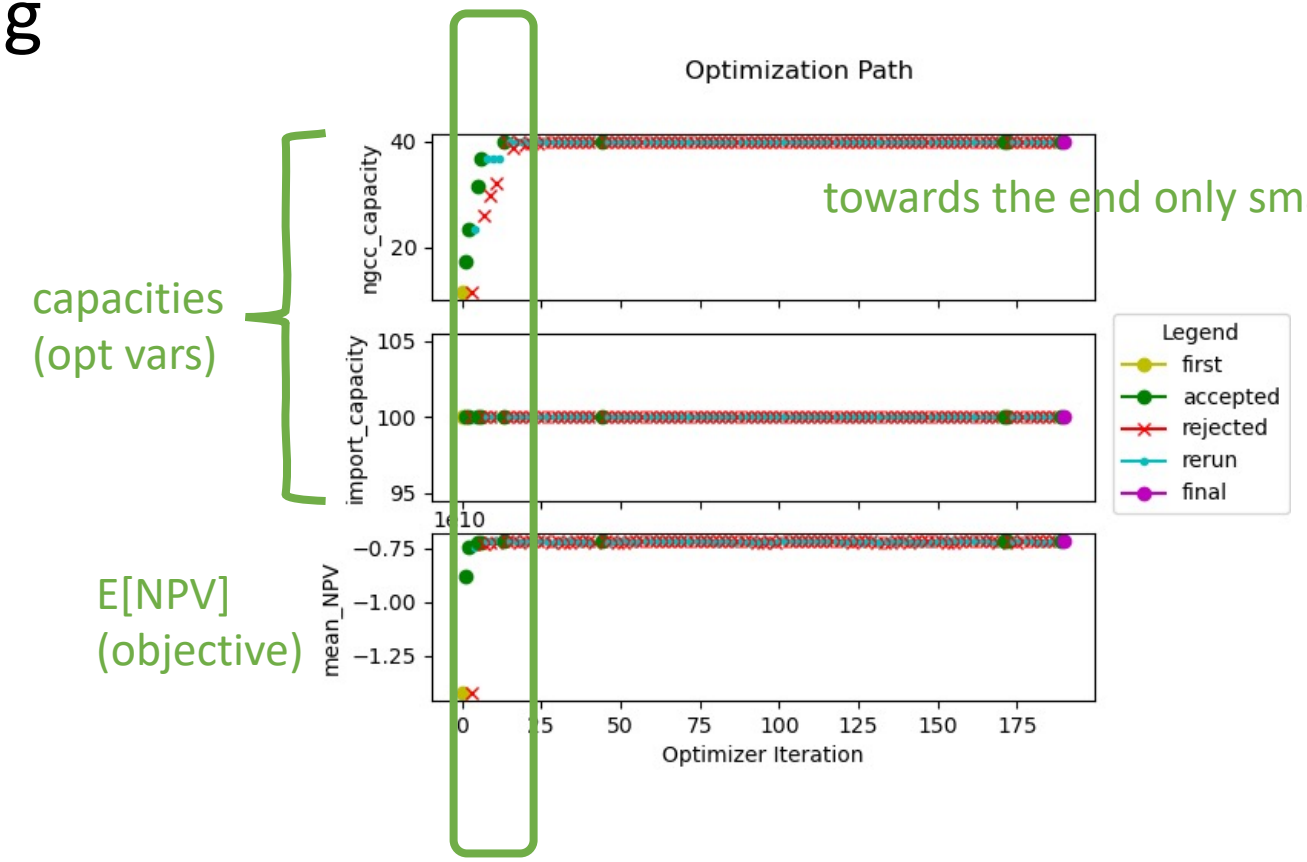
LWRS LIGHT WATER REACTOR SUSTAINABILITY

IES Integrated Energy Systems

Things to look at now that it's done

`1_simple_o/opt_path.png`

- Messy plot
- Many plots are bad
  - but some are useful
- Not intended for report quality
  - Indicative of process



Wrap up Example

- That's it!
- Takeaways:
  - 3-unit problem, single resource
  - Setting up feasible problems
  - Running and observing HERON runs
  - Viewing results

<Case>

```
<Case name="1_simple">
  <mode>opt</mode>
  <num_arma_samples>20</num_arma_samples>
  <parallel>
    <inner>2</inner>
  </parallel>
  <time_discretization>
    <year_variable>YEAR</year_variable>
    <time_variable>HOUR</time_variable>
    <end_time>23</end_time>
    <num_steps>24</num_steps>
  </time_discretization>
  <economics>
    <ProjectTime>3</ProjectTime>
    <DiscountRate>0.08</DiscountRate>
    <tax>0.1</tax>
    <inflation>0.1</inflation>
    <verbosity>50</verbosity>
  </economics>
  <dispatcher>
    <pyomo/>
  </dispatcher>
</Case>
```

<Component NGCC>

```
<Component name="ngcc">

  <produces resource="electricity" dispatch="independent">
    <capacity resource="electricity">
      <opt_bounds>10, 40</opt_bounds> <!— GW —>
    </capacity>
  </produces>

  <economics>
    <lifetime>10</lifetime>
    <!— construction cost —>
    <CashFlow name="capex" type="one-time" taxable="True" inflation="none" mult_target="False">
      <driver>
        <variable>ngcc_capacity</variable>
      </driver>
      <reference_price>
        <!— 1000 $/kW * 1e6 kW/GW = 1e9 est cost for 1 GW NGCC —>
        <fixed_value>-1e5</fixed_value>
      </reference_price>
      <!— <depreciate>15</depreciate> —>
    </CashFlow>

    <CashFlow name="var_OM" type="repeating" taxable='True' inflation='none' mult_target='False'>
      <driver>
        <activity>electricity</activity>
        <multiplier>-1</multiplier>
      </driver>
      <reference_price>
        <!— ballpark $25/MWh —>
        <fixed_value>25e3</fixed_value>
      </reference_price>
    </CashFlow>
  </economics>
</Component>
```

LWRS LIGHT WATER REACTOR SUSTAINABILITY

IES Integrated Energy Systems

<Component Import>

```
<Component name="import">

  <produces resource="electricity" dispatch="independent">
    <capacity resource="electricity">
      <fixed_value>100</fixed_value> <!— GW —>
    </capacity>
  </produces>

  <economics>
    <lifetime>1</lifetime>
    <CashFlow name="import" type="repeating" taxable='True' inflation='none' mult_target='False'>
      <driver>
        <activity>electricity</activity>
        <multiplier>-1</multiplier>
      </driver>
      <reference_price>
        <!— ballpark $100/MWh —>
        <fixed_value>100e3</fixed_value>
      </reference_price>
    </CashFlow>
  </economics>
</Component>
```

<Component Grid>

```
<Component name="grid">
  <demands resource="electricity" dispatch="fixed">
    <capacity>
      <ARMA variable="TOTALLOAD">synth</ARMA>
      <multiplier>-1</multiplier>
    </capacity>
  </demands>
  <economics>
    <lifetime>1</lifetime>
  </economics>
</Component>
```

<DataGenerators>

```
<DataGenerators>
  <ARMA name='synth' variable="TOTALLOAD">../arma_202112_nyiso_def.pk</ARMA>
</DataGenerators>
```