# Untitled

February 25, 2022

[1]:
```
from collections import defaultdict, Counter

import numpy as np
import pandas as pd
import yaml
from pyomo.environ import ConcreteModel
from pyomo.opt import SolverFactory
from powerutils.models import StochasticHurricaneMinShedLPDC
```

[2]:
```
specs = dict()
with open('/home/brent/repositories/lpdc-and-lpac/data/specs/
 ↪specs-harvey-tigerdam-disc.yaml') as fh:
    specs.update(yaml.load(fh, Loader=yaml.Loader))
with open('/home/brent/repositories/lpdc-and-lpac/data/specs/
 ↪specs-lpdc-reduced-ACTIVSg2000.yaml') as fh:
    specs.update(yaml.load(fh, Loader=yaml.Loader))
```

[3]:
```
input1 = pd.read_csv('/home/brent/Downloads/Final_Input1.csv', index_col=0)
input2 = pd.read_csv('/home/brent/Downloads/Final_Input2.csv', index_col=[0,1])
```

[4]:
```
for n, sub in input1['SubNum'].items():
    his = sub
    mine = specs['k_of_n'][n]
    if mine != his:
        print(n, mine, his)
```

[5]:
```
for n, p_gen_lo in input1['generation_capacity_min'].items():
    his = p_gen_lo / 100
    if n in specs['G_n']:
        mine = sum(specs['p_gen_lo'][g] for g in specs['G_n'][n])
    else:
        mine = 0.0
    if not np.isclose(mine, his):
        print(n, mine, his)
```

[6]:
```
for n, p_gen_hi in input1['generation_capacity_max'].items():
    his = p_gen_hi / 100
    if n in specs['G_n']:
```

```
        mine = sum(specs['p_gen_hi'][g] for g in specs['G_n'][n])
    else:
        mine = 0.0
    if not np.isclose(mine, his):
        print(n, mine, his)
```

[7]:
```
for n, load in input1['load'].items():
    his = load / 100
    if n in specs['D_n']:
        mine = sum(specs['p_load_hi'][d] for d in specs['D_n'][n])
    else:
        mine = 0.0
    if not np.isclose(mine, his):
        print(n, mine, his, len(specs['D_n'][n]))
```

```
4003 1.8026239999999998 1.2729300000000001 2
4019 1.35828 0.38253 2
4038 0.8330399999999999 0.8277899999999999 2
4042 2.15178 1.01166 2
4051 0.89391 0.15438000000000002 2
4055 0.17781 0.01608 2
4068 0.27986999999999995 0.026459999999999997 2
4111 1.0407600000000001 0.01743 2
4124 0.9662009999999999 0.03141 3
4144 0.34141699999999997 0.16356 2
4149 0.6135599999999999 0.01137 2
4155 0.21199899999999997 0.01206 3
4167 0.05307 0.023370000000000002 2
4169 0.474587 0.27681 2
4173 0.12054 0.0059099999999999995 2
4178 0.17457 0.0108 2
4186 0.37059 0.18087 2
4191 0.46434 0.015449999999999998 2
6027 3.077856 0.6497700000000001 2
6029 0.962194 0.0177 4
6096 0.721368 0.25779 2
6103 1.03356 0.02727 3
6135 0.21710999999999997 0.14658 2
6139 0.09605999999999999 0.02802 2
6189 0.27369 0.02934 3
6195 0.02655 0.009689999999999999 2
6196 1.138227 0.21276 2
6218 1.5198330000000002 0.51243 2
6253 0.13197 0.01809 2
6264 0.34968 0.2304 2
6278 3.203804 0.06813 3
6311 0.08583 0.01413 3
6327 0.1761 0.03333 2
```

```
6340 0.8609720000000001 0.16107 3
7020 0.75384 0.00378 2
7026 0.43389000000000005 0.26019 2
7129 0.05151000000000001 0.03618 2
7145 1.43784 0.13833 2
7149 0.12249 0.02154 3
7153 0.26205 0.00183 5
7174 1.24674 0.75456 2
7213 0.6850200000000001 0.27309 2
7219 0.012539999999999999 0.01023 2
7247 0.09201 0.06849 2
7248 2.20521 1.0289700000000002 2
7275 0.42306000000000005 0.00363 2
7294 0.07251 0.015629999999999998 2
7299 0.18395999999999998 0.18078 2
7302 0.24429 0.0013800000000000002 2
7306 0.07689 0.0017100000000000001 2
7330 1.9605300000000003 0.52473 2
7381 0.9475500000000001 0.44331000000000004 2
7392 0.52029 0.00936 2
7393 0.07314000000000001 0.00393 2
7404 1.7772900000000003 0.9167700000000001 2
8032 0.77961 0.038759999999999996 3
8143 1.7519850000000001 0.12666 3
```

[8]:
```
for (n, m), x in input2['BR_X'].items():
    ls = specs['L_nm'][n, m]
    if not any(np.isclose(-x, specs['b'][l]) for l in ls):
        print(n, m, -x, [specs['b'][l] for l in ls])
```

[9]:
```
for (n, m), s_flow_hi in (input2['RATE_A'] / 100).items():
    ls = specs['L_nm'][n, m]
    if not any(np.isclose(s_flow_hi, specs['s_flow_hi'][l]) for l in ls):
        if not s_flow_hi >= 400:
            print(n, m, s_flow_hi, [specs['s_flow_hi'][l] for l in ls])
```