Assignment 2: Scenario-based stochastic programming

Submitted by: Babu Kumaran Nalini

Introduction:

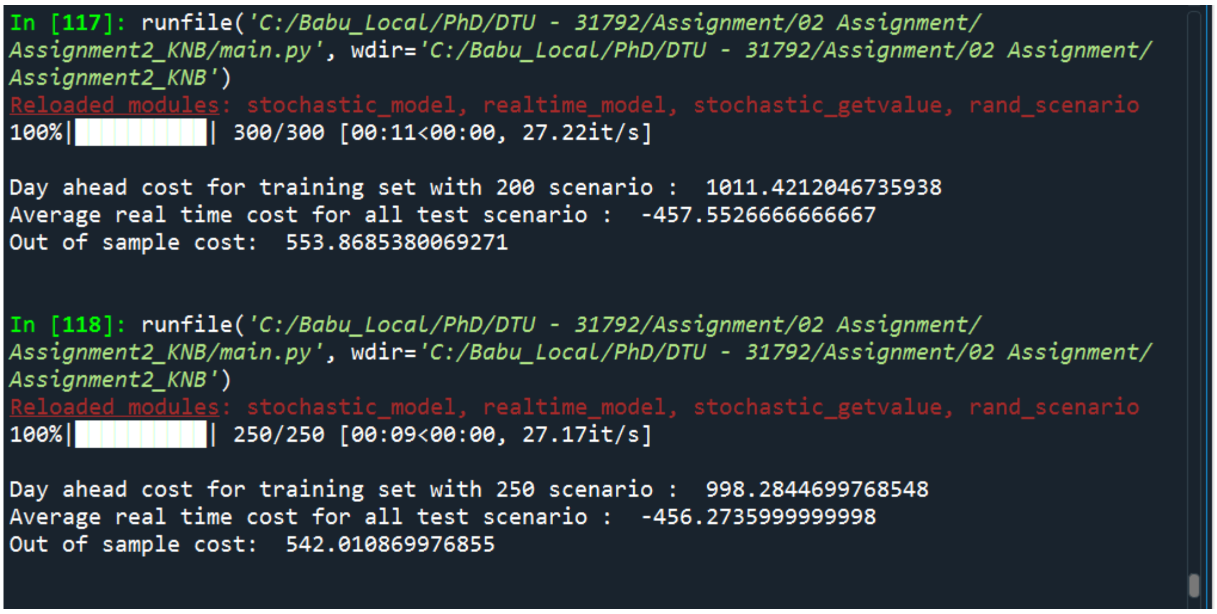
A market clearing problem is chosen for the scenario-based stochastic programming with wind power plant causing uncertainty in the system. The python code is tested in two models namely a simply example as discussed in the lecture and with the IEEE RTS 24-bus system. The result of the stochastic problem is presented as a study between the cost observed from the in-sample and out of sample analysis by dividing the scenario set into training and testing sets.

Python Model:

- Input file
  - 2-bus system (based on lecture) / 24-bus IEEE data
  - Excel data input
- Scenario size:
  - Any number of random scenarios can be generated using rand_scenario.py
  - Here, 500 random scenarios are generated added to input_data.xlsx file.
- Testing and training
  - The user can choose between the ratio of testing and training set
  - N_train sets the value of training set.
- Probability distribution
  - The probability distribution for different scenarios can be chosen between a constant or a random distribution.
  - The random distribution is done using Dirichlet function.
- Specifications
  - Scenarios: 500 (100 training + 400 testing)
  - Total constraints: >1000 for Stochastic model and >39 per real-time optimization
  - Solver: GLPK using Pyomo
  - Processor: Core i7 8550
  - Total time: 20 seconds (A progress bar is included)
- Output
  - In-sample cost
  - Average cost per real-time scenario
  - Out of sample cost

Results

In the following section, a discussion between the in-sample and out-of-sample costs are analyzed. The simple dataset is chosen here since an observable difference was seen as compared to the IEEE 24 bus dataset and the time taken to run each case for IEEE 24 bus systems is around 4 minutes. Therefore, for simplicity the 2-bus system is used below. The input file can be changed in order to run the same test for IEEE 24 bus system.



```
In [117]: runfile('C:/Babu_Local/PhD/DTU - 31792/Assignment/02 Assignment/
Assignment2_KNB/main.py', wdir='C:/Babu_Local/PhD/DTU - 31792/Assignment/02 Assignment/
Assignment2_KNB')
Reloaded modules: stochastic_model, realtime_model, stochastic_getvalue, rand_scenario
100%|██████████| 300/300 [00:11<00:00, 27.22it/s]

Day ahead cost for training set with 200 scenario :  1011.4212046735938
Average real time cost for all test scenario :  -457.5526666666667
Out of sample cost:  553.8685380069271


In [118]: runfile('C:/Babu_Local/PhD/DTU - 31792/Assignment/02 Assignment/
Assignment2_KNB/main.py', wdir='C:/Babu_Local/PhD/DTU - 31792/Assignment/02 Assignment/
Assignment2_KNB')
Reloaded modules: stochastic_model, realtime_model, stochastic_getvalue, rand_scenario
100%|██████████| 250/250 [00:09<00:00, 27.17it/s]

Day ahead cost for training set with 250 scenario :  998.2844699768548
Average real time cost for all test scenario :  -456.2735999999998
Out of sample cost:  542.010869976855
```

Figure 1 Snippet from python program execution

Figure 1 shows a snippet from the python program which results in the in-sample and out-of-sample cost from the stochastic optimization. The table 1 shows the correlation between the in-sample and out-of-sample cost with respect to the training data set size.

Table 1 Correlation between in-sample and out-of-sample cost with training set size

Training set	Test set	Probability distribution	In-sample cost	Out-of-sample cost
10	490	Equal	700	1956.9
50	450	Equal	1017.34	547.83
100	400	Equal	534.53	58.05
150	350	Equal	80	-399.3
200	300	Equal	-433.19	-907.21
250	250	Equal	-910.38	-1383.77
10	490	Dirichlet	915	547.28
50	450	Dirichlet	1055.43	594.5
100	400	Dirichlet	1006.47	552.08
150	350	Dirichlet	1029.52	560.29
200	300	Dirichlet	1011.42	553.86
250	250	Dirichlet	998.28	542.01

Discussion

- Based on the above results it is difficult to draw a conclusive pattern.
- Nevertheless, it can be found that when a Dirichlet function is used to weigh the probability of different scenarios in the stochastic model then we observe a much constant in-sample and out-sample cost.
- Equally weighing all probability basically does not improve your stochastic model and it is important to choose the probability wisely.
- It is also important to note that the initial set for the scenarios where random sample. This depreciates the possibility to find a pattern between the in-sample and out-of-sample cost as the model just uses a randomly varying scheme.
- Here there is no scenario grouping or priority allocation were performed.

****