# MATH 11247: Optimization under Uncertainty
# Assignment #3

**Due Date: March 31, 4pm**

Instructions

- This is an individual assignment.
- You **must** cite **any** references (texts, papers, websites, etc.) you have used to help you solve these problems.
- Submit **all** your solutions via Gradescope **(pdf file only)**.
- For all the implementations, please add your code and any desired screenshots of your codes' output (i.e., what you print to the screen as you run your codes to answer the assignment questions) to your written answer. (Do not submit your codes as separate files.)
- For any clarifications on the questions, feel free to post in Piazza.

---

1 [50pts] Consider the problem and instance from Question 3 of Assignment #2, copied here for your convenience:

$$\begin{aligned} \min \quad & \sum_{n \in N} \theta_n x_n + \sum_{k \in K} p_k Q_k(x) \\ \text{s.t.} \quad & \sum_{n \in N} x_n \leq I \\ & x_n \geq 0 \quad \forall n \in N \end{aligned}$$

where, for any $k \in K$,

$$\begin{aligned} Q_k(x) = \min \quad & \sum_{n \in N} \theta'_n (u_n + v_n) + h z_n + g s_n \\ \text{s.t.} \quad & I + \sum_{n \in N} u_n \geq \sum_{n \in N} v_n + \sum_{n \in N} x_n \\ & Y_n + x_n + v_n + s_n = d_n^k + z_n + u_n \qquad \forall n \in N \\ & u_n, v_n, z_n, s_n \geq 0 \qquad \forall n \in N \end{aligned}$$

A data file "data.py" and a data reader file "ReadData.py" are provided.

Now, in this assignment, implement the following algorithms to solve the instance provided via"data.py".

(a) The multi-cut Benders decomposition algorithm.

(b) The single-cut Benders decomposition algorithm.

For each algorithm, report the total time (in seconds), total number of cuts added, and number of iterations. Limit the number of iterations for each algorithm to 200. If any algorithm does not finish in 200 iterations, report the best lower bound and best upper bound obtained by the algorithm after the 200 iterations.

For each method, have your code output the upper and lower bound obtained after each iteration. Along with your source codes, provide a printout of the output of your code as it runs.

For the models you will create to solve the master linear program and the subproblem linear programs, turn off your solver's logging of progress in solving the linear program to produce a clean printout to the screen (e.g., if you are using Gurobi, you can set the the Gurobi parameters OutputFlag and/or LogToConsole parameters to 0).

2 [25pts] Consider the following problem:

$$\begin{aligned} \min \ & -0.75x + \mathbb{E}_\xi[Q(x,\xi)] \\ \text{s.t} \ & 0 \le x \le 5 \end{aligned}$$

with

$$\begin{aligned} Q(x,\xi) = \min \ & -v_1 + 3v_2 + v_3 + v_4 \\ \text{s.t.} \ & -v_1 + v_2 - v_3 + v_4 = \xi + \frac{1}{2}x \\ & -v_1 + v_2 + v_3 - v_4 = 1 + \xi + \frac{1}{4}x \\ & v_1, v_2, v_3, v_4 \ge 0 \end{aligned}$$

where $\xi$ is a random variable with Poisson distribution having the mean of $\lambda = 0.5$. For a sample size of $N = 30$ and $M = 10$ batches, provide a 95% confidence interval for the lower bound on the optimal value of the stochastic program. Among the solutions generated in these batches, choose one solution that achieves *the best sample average objective value* as the candidate solution, and using an independent sample of size $\tilde{N} = 500$, report a 95% confidence interval for the upper bound on the optimal value of the stochastic program, along with the (worst-case) optimality gap estimate.

3 [25pts] **(Quick research)** The paper [**?**] proposes a problem-driven scenario reduction approach, where a mixed-integer programming (MIP) based clustering model, given as (24)-(29) in [**?**], is solved to provide a smaller scenario set which aims to reflect a given larger sample of scenarios. Illustrate this approach on the stochastic problem given in Question 2 of this assignment: Generate a sample $\mathcal{S}$ of $N = 100$ scenarios, create and solve the proposed MIP model to obtain a smaller representative sample $\mathcal{S}'$ consisting of $N' = 10$ scenarios. Then, solve the stochastic program with the sample of scenarios $\mathcal{S}$ and with the sample of scenarios $\mathcal{S}'$ to obtain candidate first-stage solutions $x$ and $x'$, respectively. Lastly, compare $x$ and $x'$ by evaluating their true objective value by using 10,000 scenarios to approximate the expected value in the true stochastic programming model's objective.

## References

[1] Julien Keutchayan, Janosch Ortmann, and Walter Rei. Problem-driven scenario clustering in stochastic optimization. *Computational Management Science*, 20(1):13, 2023.