NATIONAL UNIVERSITY OF SINGAPORE
Department of Mathematics

2021/2022 (II) MA4260 Stochastic Operations Research Computational Assignment

Note. This assigment carries 20% of the grade for this course. You are allowed (and encouraged) to discuss this homework with your classmates. You need to write down the names of all the classmates you collaborate with.

Submission. Please submit your solutions on LumiNUS. You are expected to submit a clear write-up of the problems (about one page per problem). You are to attach a print-out of your **code**. You must write your own code. You are permitted to use any language you wish.

Deadline. 11:59pm April 8 (Friday Week 12).

1 Time-Based Simulation [5marks]

Simulation is a core part of stochastic operations research. We use simulation to model complex processes when analytical methods fail, and we use the results from simulations to understand how the processes perform.

Simulations are broadly divided between discrete event simulations and continuous simulations. For the purposes of operations research, it is more typical to work with discrete event simulations. Further within discrete event simulations, there are two basic paradigms: The first is *fixed time increments*, in which time is divided into small units. We simulate the events that occur within each time unit, and we progress. The second is *next event increment.* We progress by tracking the time of the next event.

The following exercise illustrates the first paradigm.

We operate a factory that is contracted to provide an important equipment A every week. For every week that we are unable to deliver on our contract, we pay a penalty of p.

To produce one unit of A, we need one unit of base ingredient B. Unfortunately, our only source of B is a somewhat unreliable supplier. Each week, our supplier can provide up to two units with probability 95%, each with cost c. However, there is a 5% probability that a catastrophic failure occurs on the supplier side. When that happens, our supplier will not be able to provide B for 20 *consecutive weeks*. After these 20 weeks have elapsed, we assume that the supplier will be able to provide supplies in the 21st week with probability 95%.

To mitigate the problems arising from our somewhat unreliable supplier, we keep stores of B in our inventory. Our policy is simple: In every week, we will top up as much as possible so that we have at most q units (therefore in each week we either purchase 1 or 2 units). The holding host per unit per week is h.

Our main decision variable is to estimate the optimal value $q^\star$. To do so, we will build a simulator that helps us estimate the expected total cost per unit time contributed by the holding cost and the shortage cost for different choices of q. We perform the following steps

1. Define the state by two numbers x_{inv} denoting the inventory level and x_{dis} denoting the number of days till the catastrophic failure resolves. Create an additional variable T denoting total cost (so far). At the beginning, we set $x_{inv} = q$ and $x_{dis} = 0$.

2. In each week we perform the following steps.

   (a) If $x_{dis} > 0$: This means that disaster is still on-going. Decrease $x_{dis} > 0$ by one

   $$x_{dis} \leftarrow x_{dis} - 1.$$

   Note that we are not allowed to make purchases this week, so we set the following variable as follows

   $$\texttt{PURCHASESPERMITTED} \leftarrow \texttt{False}.$$

   If $x_{dis} = 0$: This means that disaster is not occurring, so purchases are permitted, and we set

   $$\texttt{PURCHASESPERMITTED} \leftarrow \texttt{True}.$$

   (b) Next, we try to produce and provide the good if possible. If $x_{inv} > 0$: We produce the good

   $$x_{inv} \leftarrow x_{inv} - 1.$$

   If $x_{inv} = 0$, we are instead penalized for failing to deliver

   $$T \leftarrow T + p.$$

(c) If `PURCHASESPERMITTED` = `True`, that is, purchases are permitted at the start of the week, it means that the supplier may be operating. Draw a random variable $u \sim \mathsf{UNIF}[0, 1]$.

- If $u \leq 0.95$, that means purchases are permitted in this week. We purchase as much as possible so that $x_{inv} \leq q$ (supplier can provide at most 2 units). For each unit purchased, add c to the T.
- If $u > 0.95$, this means that a catastrophic failure has occurred. No purchase are permitted, and we set $x_{dis} = 19$ (number of weeks out of action).

If `PURCHASESPERMITTED` = `False`, nothing happens here.

(d) We add the holding cost to total costs

$$T \leftarrow T + hx_{inv}.$$

(e) Increment the time tracker by one week.

Repeat this procedures for large number of weeks. Plot the estimated total cost per week for different choices of q. What is the optimal $q^\star$ we should pick? Because the computed total cost is an estimate, how do you tell if your solution is quite close to the actual mean? Briefly discuss. Use the following choices of values in your response.

p	c	h
4	1	0.1
10	1	0.1
20	1	0.1

2 Event-Based Simulation [5marks]

In this problem, we create a simple simulation for describing passengers waiting at a bus stop. The exercise illustrates an instance of a *next event increment* simulation.

1. Create a random vector of dimensions 10^5 where each entry is an i.i.d. exponentially random variables with mean 1. Denote this by $x_{passenger}$. This vector represents the interarrival timings of passengers arriving at a bus-stop. Create a vector $y_{passenger}$ of equal dimensions where

$$y_{passenger,k} = \sum_{i=1}^{k} x_{passenger,i}$$

This is the actual arrival time of passenger k.

2. Create a random vector of dimensions 10^4 where each entry is an i.i.id exponentially random variables with mean 15. This represents the interarrival timings for buses. Repeat the process to generate a vector y_{bus} that describes bus arrival timings.

3. Let us assume that the bus has infinite capacity. So whenever a bus arrives, all passengers at the bus stop board the bus. (This is a bulk death scenario, and cannot be modeled by a birth-death process.) Given the two vectors $y_{passenger}$ and y_{bus}, how would you compute the wait time of the k-th passenger for a bus? Plot a histogram of the distribution of waiting timings for a random passenger for a bus.

4. How do you compute the number of passengers that the k-th bus arriving at the bus-stop picks up? Plot a histogram of the distribution of this number.

5. Notice that, based on the way we generated the arrival timings, you will very likely have a scenario where the buses at the end of the simulation did not have any passengers. Did you include the statistics for these buses in your answer to the previous question? Discuss.

3 Fitting Stochastic Models [5marks]

The basic question we investigate is as follows: Given data $\{t_1, t_2, \ldots, t_n\}$ (for instance, these could be interarrival or service timings), how do we tell if the exponential distribution is a good fit?

In this exercise, we perform a χ^2 goodness-of-fit test. (Note that computer programming is not necessary for this problem.) We perform the following hypothesis test:

1. H_0: $\{t_1, t_2, \ldots, t_n\}$ are samples drawn from a random exponential random variable with parameter λ.

The χ^2 goodness-of-fit test applies to general distributions. Here, we describe the test specialized to the exponential variable.

Let $\hat{\lambda}$ be an estimate of the parameter λ given as follows

$$\hat{\lambda} = \frac{n}{\sum_{i=1}^{n} t_i}.$$

We define the density function as follows

$$f(t) = \hat{\lambda} e^{-\hat{\lambda} t}, \quad t \geq 0.$$

Choose the number of bins k. Set $\beta_0 = 0$ and $\beta_k = \infty$, and partition the support of the random variable $[\beta_0 = 0, \beta_k = \infty)$ into k sub-intervals $[\beta_0, \beta_1)$, $[\beta_1, \beta_2), \ldots, [\beta_{k-1}, \beta_k)$. Let

$$e_i = n \int_{\beta_{i-1}}^{\beta_i} f(t) dt \quad i = 1, 2, \ldots, k.$$

That is to say, e_i is the expected number of times we expect the random variable to be in the bin $[\beta_{i-1}, \beta_i)$.

Earlier, one way of picking β_i is so that $e_i/n = 1/k$ for all i. This is equal to saying that the random variable has equal probability being in $[\beta_i, \beta_{i+1})$.

Let O_i = number of t_j's which fall in $[\beta_{i-1}, \beta_i)$. Define the observed value of the chi-square statistic

$$\chi^2(obs) = \sum_{i=1}^{k} \frac{(O_i - e_i)^2}{e_i}$$

The value of $\chi^2(obs)$ follows a chi-square distribution χ^2_ν, with "$\nu = k - r - 1$" degree of freedom, where $r = 1$ if $f(t)$ is an exponential distribution, $r = 2$ if $f(t)$ is a normal distribution or an Erlang distribution.

If $\chi^2(obs)$ is small, we accept the null H_0; otherwise we reject H_0.

For any $\alpha \in (0, 1)$, we define a percentile point $\chi^2_\nu(\alpha)$ by

$$\alpha = \mathbb{P}(\chi^2_\nu > \chi^2_\nu(\alpha))$$

We do not reject H_0 if $\chi^2(obs) \leq \chi^2_{k-r-1}(\alpha)$ and reject H_0 otherwise. A smaller choice of α makes it easier to accept the null H_0. (A χ^2 table can be found in the assignment folder.)

Problem

Consider the following service times (in minutes):

0.01,	0.07,	0.03,	0.08,	0.04
0.10,	0.05,	0.10,	0.11,	0.17
1.50,	0.93,	0.54,	0.19,	0.22
0.36,	0.27,	0.46,	0.51,	0.11
0.56,	0.72,	0.29,	0.04,	0.73

Does it seem reasonable to conclude that these observation come from an exponential distribution? Perform a χ^2 test with $\alpha = 0.5$.

4 Stochastic Programming [5marks]

The goal of this example is to get you acquainted with using software to solve your optimization problems. This involves setting up the appropriate solver for your needs, and formulating optimization problems into the appropriate format for your solver.

There is a huge list of available optimization software. Popular choices include

1. CPLEX `https://www.ibm.com/analytics/cplex-optimizer`
2. Gurobi `https://www.gurobi.com/`

These solvers are commercial and can solve a fairly generic class of problems. These can be used within most popular programming languages including MATLAB, Python, Java, C++, ...

To set up Gurobi to run on your Python distribution, you can use the following command:

```
pip install -i https://pypi.gurobi.com gurobipy
```

Note that this command only installs a limited but free copy. You can get a free academic license by following instructions on the website. In the assignment folder, you will find the solution to the Farmer's example in our lectures in the file coded up using the Gurobipy interface in `FarmerExample.ipynb`.

An alternative to running Python on your local machine is to use Google's Colab function. This allows you to run Python codes remotely. All you need is a Google account and a web-browser. Look for `FarmerExample.html` within the homework folder, and run all the commands as it is.

For this assignment: Solve the extensive form of the Farmer's example but with the probability of above average, average and below average yields equal to $(0.5, 0.0, 0.5)$ respectively. Give the optimal solution and the optimal allocation.

Hint: To model integer variables in Gurobipy, you may use the following command

```
model.addVars(vtype = GRB.INTEGER, name="name")
```

If you wish to model binary variables instead, change `GRB.INTEGER` to `GRB.BINARY`.