Detailed Report for Assignment 2, Question 2

This document explains how our Python script fulfills **Question 2** from the DAMO600 Prescriptive Analytics assignment. All cost formulas are displayed as separate LaTeX blocks per your request.

1. Question 2 Requirements

The assignment specifies that the Python script must:

- Simulate two fan replacement policies via Monte Carlo.
- Perform exactly 45 replacement events per simulation run.
- Randomly generate both fan lifetimes and technician arrival delays.
- Calculate downtime cost and labor cost for each event.
- Sum these costs to produce a total cost per policy per run □filecite□turn7file4□.

2. Simulation Overview

The simulation compares two strategies:

1. **Current Policy**: Replace only the failed fan each time.
2. **Proposed Policy**: Replace all three fans whenever any one fails.

Each failure event incurs three cost components, calculated as follows.

Replacement Cost

$$ C_{\mathrm{rep}} ,=, n \times c_{\mathrm{fan}} $$

where

$$ \begin{aligned} n &= \text{number of fans replaced (1 or 3)},\ c_{\mathrm{fan}} &= $32. \end{aligned} $$

Downtime Cost

$$ C_{\mathrm{down}} ,=, \bigl(D + T_{\mathrm{rep}}\bigr) \times c_{\mathrm{down}} $$

where

$$ \begin{aligned} D &= \text{technician arrival delay (minutes)},\ T_{\mathrm{rep}} &= \text{replacement duration (20 min for current; 40 min for proposed)},\ c_{\mathrm{down}} &= $10;/\mathrm{min}. \end{aligned} $$

Labor Cost

$$ C_{\mathrm{lab}} ,=, \frac{T_{\mathrm{rep}}}{60} \times c_{\mathrm{labor}} $$

where

$$ \begin{aligned} c_{\mathrm{labor}} &= $30;/\mathrm{hr}. \end{aligned} $$

The total cost for one failure event is

$$ C_{\mathrm{event}} = C_{\mathrm{rep}} + C_{\mathrm{down}} + C_{\mathrm{lab}}. $$

A full simulation run of $N = 45$ failures yields

$$ C_{\mathrm{total}} = \sum_{i=1}^{45} C_{\mathrm{event},i}. $$

3. Random Sampling of Inputs

We sample from the discrete distributions given in the assignment PDF:

- **Fan Lifetimes (hours)** { 1000, 1100, 1200, 1300, 1400, 1500, 1600, 1700, 1800, 1900 } with probabilities {0.10, 0.13, 0.25, 0.13, 0.09, 0.12, 0.02, 0.06, 0.05, 0.05}:

```
lifetime = np.random.choice(fan_lifetimes, p=fan_probs)
```

- **Technician Delays (minutes)** {20, 30, 45} with probabilities {0.60, 0.30, 0.10}:

```
delay = np.random.choice(delay_times, p=delay_probs)
```

4. Core Simulation Function

We encapsulate each policy’s event loop in a function:

```
def simulate_policy(fans_to_replace, rep_time):
    total_cost = 0.0
    for _ in range(num_failures):
        lifetime = np.random.choice(fan_lifetimes, p=fan_probs)
        delay    = np.random.choice(delay_times, p=delay_probs)
        C_rep    = fans_to_replace * fan_unit_cost
        C_down   = (delay + rep_time) * downtime_cost_per_min
        C_lab    = (rep_time / 60.0) * labor_cost_per_hour
        total_cost += (C_rep + C_down + C_lab)
    return total_cost
```

This is invoked twice per run:

```
cost_current  = simulate_policy(1, 20)
cost_proposed = simulate_policy(3, 40)
```

5. Monte Carlo Loop and Data Capture

A fixed seed ensures reproducibility. We execute 1 000 runs, collecting totals:

```
np.random.seed(42)
results = []
for _ in range(1000):
    results.append((
        simulate_policy(1, rep_time_current),
        simulate_policy(3, rep_time_proposed)
    ))

df_results = pd.DataFrame(
    results,
    columns=["Cost_Current_Policy", "Cost_Proposed_Policy"]
)
```

6. Visualization of Outcomes

We visualize `df_results` via histogram and boxplot to compare distributions of total costs across policies.

7. Manual Verification Example

Fixing delay $D = 20$ min yields per-event costs:

$$ \begin{aligned} \text{Current:}\quad &C_{\mathrm{rep}}=1\times32=32,\quad C_{\mathrm{down}}=(20+20)\times10=400,\ & C_{\mathrm{lab}}=(20/60)\times30=10,\quad C_{\mathrm{event}}=32+400+10=442.\ \text{Proposed:}\quad &C_{\mathrm{rep}}=3\times32=96,\quad C_{\mathrm{down}}=(20+40)\times10=600,\ & C_{\mathrm{lab}}=(40/60)\times30=20,\quad C_{\mathrm{event}}=96+600+20=716. \end{aligned} $$

Aggregating 45 failures:

$$ 45\times442=19{,}890,\quad 45\times716=32{,}220. $$

Comparing these hand-computed totals to Monte Carlo means confirms correctness.

8. Conclusion and Recommendation

We compute average total cost per policy over 1 000 runs; the policy with the lower mean is recommended.

References DAMO600 P25 Assignment 2 Description filecite turn7file4