EE 214 Machine Problem 1.2

Paul Vincent S. Nonat
Electrical and Electronics Engineering Institute
University of the Philippines Diliman
paul.vincent.nonat@eee.upd.edu.ph

1. The Problem

A particular system in operation has 100 components. Each component has a failure probability q independent of the other components. The system is operational if any one of conditions a and b is satisfied:

a) Components 1,2,...,40 all work, and any one of components 41,42,...,50 works.

b) All of components 51,52,...,80 works and any one of components 81,82,...,100 works.

1.1) What is the probability that the system is operational? Write function *system operational()* in Python 3 code that computes for the system probability.
1.2) Suppose we can replace any 20 of these components with an equivalent no, of ultra-reliable components, each of which has a failure probability of $q/10$. Which components must be prioritized in the replacement to increase system reliability?
1.3) Suppose the failure probability of components in a is better than the components in b. That is, $q_a = 0.67q_b$ where q_a and q_b are the failure probabilities of each of the components in a and b, respectively. Determine which components should be replaced with the ultra-reliable components $0.20q_a$ at minimum cost.

Question 1.2 and 1.3 are independent of each other. Solve 1.1, 1.2, and 1.3 using simulation with $q = q_a = 0.275$. For each replacement of a regular component, is it necessary to perform 100 trials? Are 100 trials sufficient to conclude which component should be replaced?

2. The Solution

This section is divided into three parts. The first section tackles the method to determine the operational probability of the whole system. The second is focused on deciding which 20 components must be replaced, and the later focus on choosing the component replacement at the minimum cost.

2.1. *Operational Probability*

According to the problem, the system is functional if we were able to satisfy any one of the given conditions: sub-system a works or sub-system b works. It indicates that the system is composed of two groups, group a and b connected in parallel with one another, as shown in Fig. 1. Since these components are identical, we can apply the concept of independent trials to determine if the system is operational or not.

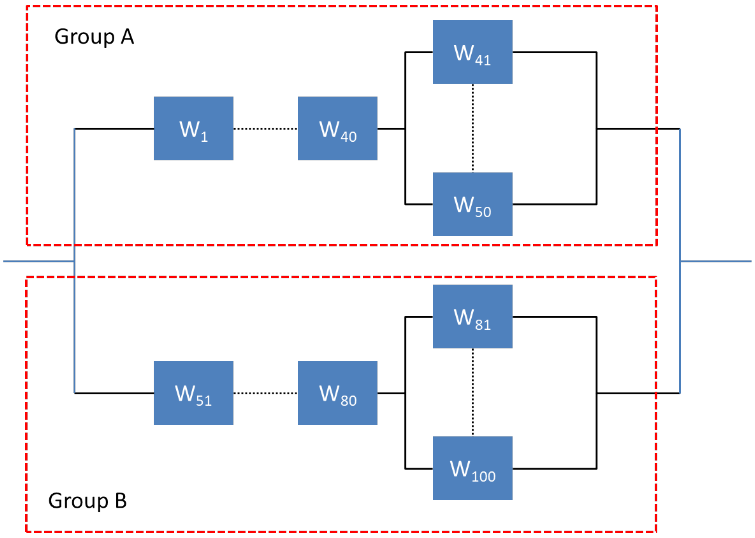

Fig. 1. System Configuration

The probability that a component is operational is given by

$$P[W_i] = 1 - q \tag{1}$$

$$i = 1, 2, ..., 100 \tag{2}$$

For components connected in series, the operational probability $P[W_s]$ is given by

$$P[W_s] = p^n \tag{3}$$

$$P[W_s] = (1 - q)^n \tag{4}$$

where n denotes the number of components connected in series. On the other-hand, components connected in parallel has the following operational probability

$$P[W_p^c] = q^n \tag{5}$$

$$P[W_p] = 1 - P[W_p^c] \tag{6}$$

In which $P[W_p^c]$ is the failure probability for components connected in parallel and $P[W_p]$ for working probability.

Applying this principle to the problem, we get

$$P[W_A] = (1-q)^{40}[1-q^{10}] \tag{7}$$

$$P[W_B] = (1-q)^{30}[1-q^{20}] \tag{8}$$

Where $P[W_A]$ and $P[W_B]$ is the working probability for system a and b. The first term of these equations pertain to the number of components in series, and the latter for parallel. The operational probability $P[W]$ of the system is as follows:

$$P[W] = 1 - (1 - P[W_A])(1 - P[W_B]) \tag{9}$$

Since group a and b is connected in parallel with one another.

2.2. *Replacement Priority for twenty components*

Replacing system components with ultra-reliable ones will allow the system to have higher operational probability, which is true if we were able to select the best component to replace since the system is configured in a series and parallel manner. According to eqn. 7 and 8, we can observe that the system has 4 subgroups. These are $P[W_{As}] = (1-q)^{40}$, $P[W_{Ap}] = [1-q^{10}]$, $P[W_{Bs}] = (1-q)^{30}$,and $P[W_{Bp}] = [1-q^{20}]$.

By definition of series and parallel components as presented in eqn. 1 and eqn. 4, we can see that components connected in series have lower operational probability compared to those in parallel. However, in our case, the parallel components on each group are connected in series with the other. Therefore, the component replacement must prioritize those who are connected in series for both groups.

Given that group a and b are connected in parallel with one another. This case suggests that increasing the operational probability for either one of them will increase the overall functional probability of the whole system. The replacement must be done as follows

$$P[W_A] = (1-q)^{40-w}(1-q)^w[1-q^{10-x}q^x] \tag{10}$$

$$P[W_B] = (1-q)^{30-y}(1-q)^y[1-q^{20-z}q^z] \tag{11}$$

Where

$$w + x + y + z = 20 \tag{12}$$

In which w,x,y, and z denotes the components replaced in $P[W_{As}]$, $P[W_{Ap}]$, $P[W_{Bs}]$,and $P[W_{Bp}]$.By inspection on eqn. 7 and eqn. 8, we can see that $P[W_B] >$

$P[W_A]$ since group b has less number of components connected in series. Hence, we can reduce the calculation time by focusing only on $P[W_B]$.

$$P[W_B] = (1-q)^{30-y}(1-q)^y[1-q^{20-z}q^z] \tag{13}$$

$$y + z = 20 \tag{14}$$

In here, the scope of determining which component to replace was reduced to those exclusively part of sub-group $P[W_{Bs}]$ and $P[W_{Bp}]$.

2.3. *Component Replacement at minimum cost*

The 3rd problem is solved independently from the second problem. In here, the components in system a is better than b that is $q_a = 0.67q_b$. To determine the components to be replaced, we used a similar approach as eqn. 10 and eqn. 11.

$$P[W_A] = (1-q)^{40-w}(1-(\frac{q}{10})^w[1-q^{10-x}(\frac{q}{10})^x] \tag{15}$$

$$P[W_B] = (1-q)^{30-y}(\frac{q}{10})^y[1-q^{20-z}(\frac{q}{10})^z] \tag{16}$$

$$w + x + y + z = n \tag{17}$$

However, in this case, the total number of components to be replaced is equal to n, which is the minimum number of components to be replaced to make the operational probability $P[W_n] > P[W]$.

3. Results and Discussion

This section is discussed as follows: System operational probability, component replacements, and minimum component replacement.

3.1. *System Operational Probability Result*

To compute for system probability, a python program was implemented with the following functions in python.

```
def series1_40():
    prob_series=pow((1-q),40)
    return prob_series;

def parallel41_50():
    prob_parallel=1-pow(q,(10))
    return prob_parallel;

def series51_80():
    prob_series=pow((1-q),30)
```

```
    return prob_series;

def parallel81_100():
    prob_parallel=1-pow(q,(20))
    return prob_parallel;

def system_operational(components_failure):
    p_a=series1_40()*parallel41_50()
    p_b=series51_80()*parallel81_100()
    p_system=1-((1-p_a)*(1-p_b))
    return p_system;
```

These functions implement the equations discussed in section 2.1. The system operational function calculates the system operational probability, as discussed in eqn. 7-9, which returns the operational probability of the system at $P[W] = 6.717741055606297e - 05$.

3.2. *Twenty Component Replacement Result*

To accurately determine the components to be replaced, two methods were used. The first method used was the one mentioned in eqn. 10-12, which considers all the components and their subgroup $P[W_{As}]$,$P[W_{Ap}]$,$P[W_{Bs}]$,$P[W_{Bp}]$. A simple modification was done on the previous python function introduced in 3.1.

```
def series1_40(w):
    prob_series=pow((1-q),40-w)*pow((1-(q/10)),w)
    return prob_series;

def parallel41_50(x):
    prob_parallel=1-pow(q,(10-x))*pow((q/10),x)
    return prob_parallel;

def component_replacementlist():
        replaced_components = list()
        len([replaced_components.append((a,b,c,d)) for a in range(21)
        for b in range(21) for c in range(21)
        for d in range (21) if a + b + c+d == 20 and b<=10])
        return replaced_components;

def series51_80(y):
    prob_series=pow((1-q),30-y)*pow((1-(q/10)),y)
    return prob_series;

def parallel81_100(z):
```

```
    prob_parallel=1-pow(q,(20-z))*pow((q/10),z)
    return prob_parallel;
```

These functions now require parameters w, x, y, z as presented in eqn. 10-12. The *component replacementlist* function, on the other hand, determines all the possible replacement combinations. The operational probability is then calculated on every combination using the *system operational* function presented in the previous section.

Table 1. All Component Replacement Result

Components	Number of Components Replaced
Component 1-40	0
Component 41-50	0
Component 51-80	20
Component 81-100	0

Note: Total Number of trials performed: 1551
max $P[W] = 0.022973127892022105$

Using this method, a total of **1551 trials** are needed for all possible combinations to determine which component to replace, as presented in Table 1. A quicker way for this is by using eqn. 13-14, since it was apparent that $P[W_B] > P[W_A]$. Implementing this method will only require modification for functions in the system b. Hence, only the combination for $y + z = 20$, as presented in the code below is needed to be computed, which is way smaller than the previous method that searches through all possible component combinations.

```
def series51_80(y):
    prob_series=pow((1-q),30-y)*pow((1-(q/10)),y)
    return prob_series;

def parallel81_100(z):
    prob_parallel=1-pow(q,(20-z))*pow((q/10),z)
    return prob_parallel;

def component_replacementlist():
        print("Generating Replacement Combinations")
        replaced_components = list()
        len([replaced_components.append((c,d)) for c in range(21)
        for d in range (21) if c+d == 20])

        return replaced_components;
```

By taking this method, the trials return the same operational probability and combinations of components to be replaced, as shown in Table 2, at a lower number of tests and faster simulation.

Table 2. Replace Components only in b

Components	Number of Components Replaced
Component 1-40	0
Component 41-50	0
Component 51-80	20
Component 81-100	0

Note: Total Number of trials performed: 21
max $P[W] = 0.022973127892022105$

3.3. *Minimum Component Replacement Result*

In determining the minimum component replacements, the series and parallel functions in the python program were modified to implement eqn. 15-17.

```
def series1_40(w):
    prob_series=pow((1-qa),40-w)*pow((1-(0.20*qa)),w)
    return prob_series;

def parallel41_50(x):
    prob_parallel=1-pow(qa,(10-x))*pow((0.20*qa),x)
    return prob_parallel;

def series51_80(y):
    prob_series=pow((1-(qa/0.67)),30-y)*pow((1-(0.20*qa)),y)
    return prob_series;

def parallel81_100(z):
    prob_parallel=1-pow((qa/0.67),(20-z))*pow((0.20*qa),z)
    return prob_parallel;

def component_replacementlist(n):
        print("Generating Replacement Combinations for n="+str(n))
        replaced_components = list()
        len([replaced_components.append((a,b,c,d)) for a in range(41)
        for b in range(11) for c in range(31)
        for d in range (21) if a + b + c+d == n])

        return replaced_components;
```

```
for i in range(101):
    replacements.append(component_replacementlist(i))

print("Calculating All possible working probability")

results=list()
for n in range(101):
    result_list= list()
    for x in range(len(replacements[n])):
       result_list.append(system_operational(replacements[n][x][0],
       replacements[n][x][1],replacements[n][x][2],replacements[n][x][3]))

    results.append(result_list)
```

The series and parallel function that calculates the series and parallel probabilities for components in a and b was modified similar to 3.2 But instead of using $\frac{q}{10}$, $0.20q_a$ was used instead. In terms of determining the combination for w,x,y, and$z$, the program prioritize those components connected in series before replacing components in parallel as n increases. The simulation produces the following output presented in Fig. 2.

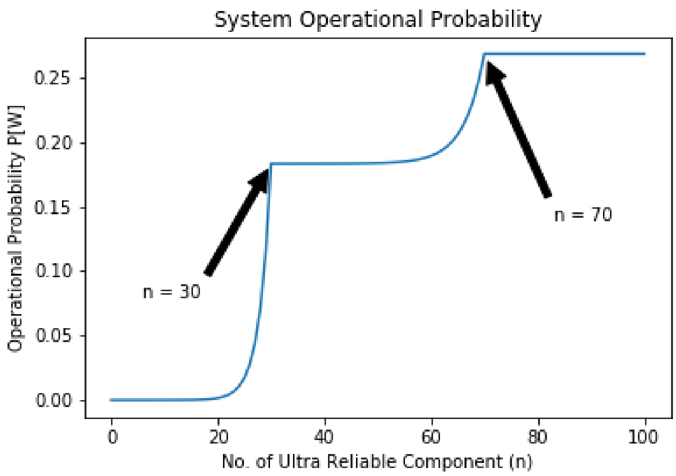

Fig. 2. Operational Probability for n replacements.

Before doing the replacement, the operational probability for components in a $P[W_A]$ is greater than those in b. However, upon doing a component replacement, the sub-system b was able to catch up, given that it has ten fewer components in series. Based on the simulation result, only two possible values of n (i.e $n = 30$

and $n = 40$) is notable. However, the difference between the probability between the two points is not that far. Hence it is more economical to replace only the **30 series components** under the system b to have a better operational probability.

4. Conclusion

The operational probability of components connected into two groups of series and parallel network sub-system a and b were able to determine using the principles of independent trials. For the problem of replacing 20 components with an ultra-reliable version, it has been concluded that it is needed to prioritize the replacement components connected in series of system b to increase the operational probability of the system since $P[W_A] > P[W_B]$ on the original configuration.

For the case of components in the sub-system, a having a better operational probability than those in sub-system b, in which we were tasked to determine which elements to be replaced with ultra-reliable versions at minimum cost. The simulation result shows that replacing all 30 series components in sub-system b will improve the operational probability of the system at a minimum cost. This option was decided given that the next increase was to change the latter series components in a, which will dramatically increase the number of ultra-reliable components used to 70 for a small reliability increase. This option is costly, hence replacing only 30 elements in series of sub-system b was the most economical option.

In conducting component replacement trials, 100 trials are sufficient due to the following reason: The first case of replacing only twenty components shows that it is not necessary to check all replacement combinations of the system. Instead, the replacement needs to be focused solely on sub-system b since it is higher than a. For the last replacement case, 100 trials are enough to test all component replacement cases from $1 \leq n \leq 100$.