Reliability Design

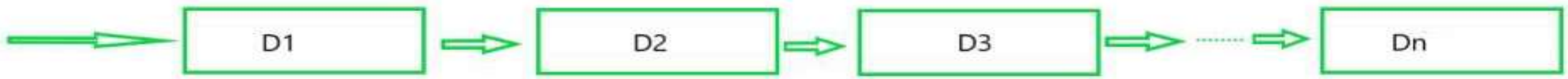
By: Rasad Regmi

Introduction to Reliability Design

Reliability Design Problem In reliability design, the problem is to design a system that is composed of several devices connected in series.

Reliability design focuses on ensuring that a system performs its intended function without failure over a specified period.

Reliable systems minimize downtime and maintenance costs, ensuring consistent performance.



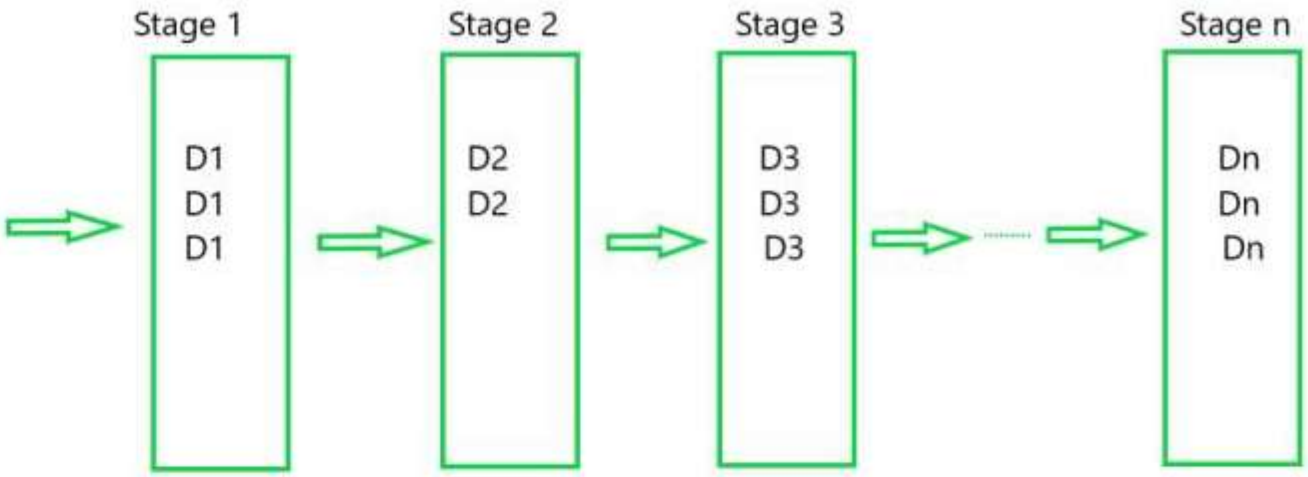
n devices D_i, 1<=i<=n, connected in series

Let r_i be the reliability of device Di (i.e. r_i is the probability that device i will function properly) then the reliability of the entire system is πr_i.

Even if the individual devices are very reliable, the reliability of the system may not be very good.

For example, if n=10 and $r_i = 0.99$, $1 \leq i \leq 10$, then $\pi r_i = 0.904$.

Hence, it is desirable to duplicate devices. Multiple copies of the same device type are connected in parallel.



n devices D_i, 1<=i<=n, connected in series

If stage i contains mi copies of device D_i. Then the probability that all m_i have a malfunction is $(1 - r_i)^{m_i}$. Hence the reliability of stage i becomes $1 - (1 - r_i)^{m_i}$.

The reliability of stage 'i' is given by a function $\phi_i(m_i)$.

Our problem is to use device duplication. This maximization is to be carried out under a cost constraint. Let c_i be the cost of each unit of device i and let c be the maximum allowable cost of the system being designed.

Maximum number of copies for D_i

Assume each $C_i > 0$, each m_i must be in the range $1 \le m_i \le u_i$, where

$$u_i = \left\lfloor \left(C + C_i - \sum_{1}^{n} C_j \right) \Big/ C_i \right\rfloor$$

Dominance Rule

In reliability design, the dominance rule is a principle used to optimize system reliability by comparing different configurations or components.

The dominance rule $(f1, x1)$ dominate $(f2, x2)$ if $f1 \geq f2$ and $x1 \leq x2$. Hence, dominated tuples can be discarded from S^i (where i denotes stage number).

Code

reliabilityDesign.cpp X

reliabilityDesign.cpp > main()

```cpp
#include <iostream>
#include <cmath>

using namespace std;

// Calculating system reliability with the given number of copies
double calculateSystemReliability(double R, int m)
{
    return 1 - pow(1 - R, m);
}
```

reliabilityDesign.cpp

reliabilityDesign.cpp > main()

```cpp
int main()
{
    double C1, R1, C2, R2, C;
    int u1, u2;

    cout << "Enter the cost of device 1 (C1): ";
    cin >> C1;
    cout << "Enter the reliability of device 1 (R1): ";
    cin >> R1;

    cout << "Enter the cost of device 2 (C2): ";
    cin >> C2;
    cout << "Enter the reliability of device 2 (R2): ";
    cin >> R2;

    cout << "Enter the total available cost (C): ";
    cin >> C;

    // Calculating max no. of copies for 2 devices
    u1 = (C + C1 - (C1 + C2)) / C1;
    u2 = (C + C2 - (C1 + C2)) / C2;
```

reliabilityDesign.cpp ×

reliabilityDesign.cpp > main()

```cpp
int main()
    double bestReliability = 0.0, bestCost = 0.0;
    int bestM1 = 0, bestM2 = 0;

    for (int m1 = 1; m1 <= u1; m1++)
    {
        for (int m2 = 1; m2 <= u2; m2++)
        {
            double totalCost = m1 * C1 + m2 * C2;
            if (totalCost <= C)
            {
                double currentReliability = calculateSystemReliability(R1, m1) * calculateSystemReliability(R2, m2);
                if (currentReliability > bestReliability)
                {
                    bestReliability = currentReliability;
                    bestM1 = m1;
                    bestM2 = m2;
                    bestCost = totalCost;
                }
            }
        }
    }

    cout << "\nThe best design has a reliability of " << bestReliability << endl;
    cout << "Total cost of the best design is " << bestCost << endl;
    cout << "Tracing back for the solution, we can determine that m1=" << bestM1 << " and m2=" << bestM2 << "\n\n";

    return 0;
}
```


THANK YOU!