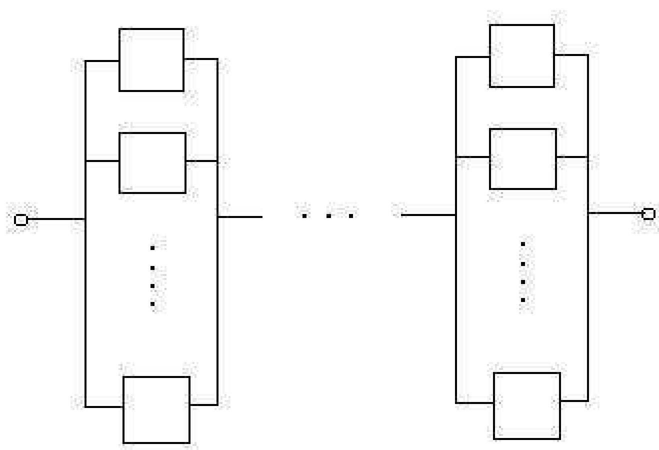
Combined Series and Parallel Configurations

Series-parallel configuration

The general series-parallel configuration consists of N subnetworks, each is a parallel configuration, containing M parallel components. The subnetworks are connected in series. The network reliability can be computed using the known formulas for the series and parallel configurations. If each component has reliability p, we obtain

$$R = [1 - (1 - p)^M]^N,$$

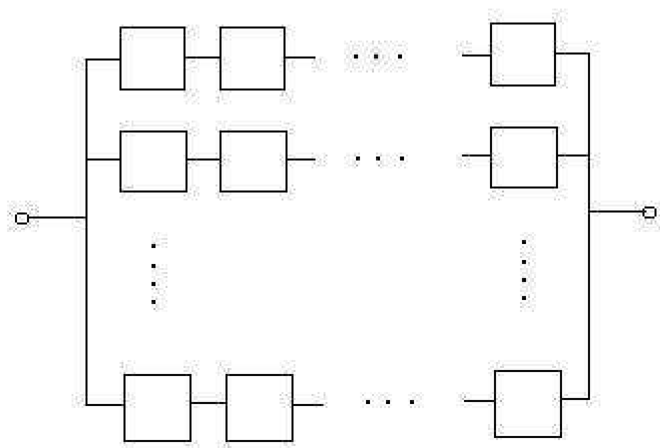


Parallel-series configuration

In the parallel-series configuration M series subnetworks are connected in parallel. Each series subnetwork contains N components. It follows again directly from the known formulas that the network reliability can be computed as

$$R = 1 - (1 - p^N)^M$$

if each component has reliability p.



Exercise

Comparison of the series-parallel and parallel-series configurations:

Assume $N = M$, that is, we have the same number of components in each column and row. Also assume that each component has the same reliability p with $0 < p < 1$. Which of the two configurations tends to be more reliable if N grows very large ($N \to \infty$)?

Solution

With $N = M$ we obtain that the series-parallel configuration has reliability

$$R_{s-p} = [1 - (1-p)^N]^N,$$

and the parallel-series has reliability

$$R_{p-s} = 1 - (1 - p^N)^N$$

Which of these similarly looking expressions is larger if N tends to infinity?

We can answer the question using the following known formula from calculus:

$$\lim_{x \to 0} (1-x)^{1/x} = e^{-1}$$

where $e = 2.71...$ is the base of the natural logarithm. We use it for approximation: when x is very small, then

$$(1-x)^{1/x} \approx e^{-1}$$

holds.

For R_{s-p} let us take $x = (1-p)^N$. Then we have $x \to 0$ if $N \to \infty$. We can then write

$$R_{s-p} = (1-x)^N = (1-x)^{(1/x)Nx} = [(1-x)^{1/x}]^{Nx} \approx e^{-Nx},$$

using $(1-x)^{1/x} \approx e^{-1}$ in the last step. Since $Nx = N(1-p)^N$, therefore, we have $Nx \to 0$, as the second factor tends to 0 exponentially, while the first grows only linearly. Thus, we obtain that

$$R_{s-p} \to 1 \tag{1}$$

as N grows.

For R_{p-s} we take $x = p^N$. Then, similarly to the above, we have $x \to 0$ if $N \to \infty$, so we can write

$$R_{p-s} = 1 - (1-x)^N = 1 - (1-x)^{(1/x)Nx} = 1 - \left[(1-x)^{1/x}\right]^{Nx} \approx 1 - e^{-Nx}.$$

Since now $Nx = Np^N$, therefore, we have again $Nx \to 0$, as the second factor tends to 0 exponentially, while the first grows only linearly. Thus, we obtain that $e^{-Nx} \to 0$, implying

$$R_{p-s} \to 0 \tag{2}$$

Comparing (1) and (2), we can conclude that with $N \to \infty$ the reliability of the series-parallel configuration tends to 1, while the reliability of the parallel-series configuration tends to 0. Thus, the series-parallel configuration will be more reliable when N grows very large.

Interpretation

It is interesting that the two apparently similar configurations behave so differently. The series-parallel tends to be more reliable as its size grows, while the parallel-series loses reliability with growing size. How could we informally explain this difference?

The series-parallel configuration has much more possible paths between the endpoints. We can pick any component from each parallel subnetwork to build a path, yielding N^N possible paths. On the other hand, the parallel-series configuration has only N possible paths between the endpoints, as each series subnetwork can serve as such a path, but there is no more possibility.

Similarly, if we consider minimum cuts, that is, minimal subsets of components whose failure disconnects the network, then we find that the situation is reversed. The series-parallel configuration has only N possible minimum cuts (taking an entire parallel subnetwork), while the parallel-series configuration has N^N minimum cuts (taking one component from each series subnetwork).

The above considerations show that the series-parallel configuration is much more connected, i.e., it has much more paths and much less cuts than the other. Thus, it is not surprising that it grows more reliable.