Proceedings of the 33rd European Safety and Reliability Conference (ESREL 2023)
Edited by Mário P. Brito, Terje Aven, Piero Baraldi, Marko Čepin and Enrico Zio
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-18-8071-1_P274-cd

A new efficient method to find a suboptimal allocation of components in a series-parallel structure

Jacek Malinowski

Systems Research Institute, Polish Academy of Sciences, Poland. E-mail: jacek.malinowski@ibspan.waw.pl

A method of optimizing the allocation of two-state independent components in a series-parallel structure is presented. Such a structure is composed of parallel substructures arranged in series and is considered operable if at least one component in each substructure is operable. The components are assumed to have different failure probabilities, so the system reliability depends on where they are located in the structure. The optimal allocation minimizes the system failure probability or, alternatively, maximizes its reliability. Interestingly enough, while the optimal components allocation problem for a parallel-series structure (i.e. series substructures arranged in parallel) has a well-known simple solution, the same does not hold for a series-parallel one. The considered problem has been investigated by several researchers who proposed quite elaborate solutions. This paper presents a recently developed, simple and efficient procedure finding a nearly optimal allocation, and sometimes the optimal one. The presented approach is based on a theorem specifying a threshold value that cannot be exceeded by a series-parallel system's reliability. Starting from some random allocation and using pairwise interchanges of components between the parallel substructures, the algorithm finds successive allocations that yield system reliabilities oscillating towards the value specified by the above theorem. In this way a suboptimal (or, many a time, optimal) allocation is obtained. An important feature is that the method's accuracy is expressed by the easy-to-compute upper bound of the difference between the optimal reliability and the obtained suboptimal value. The performed tests show that the method allows to find a (sub)optimal solution in a relatively small number of steps. Illustrative examples are given that demonstrate the method's modus operandi.

Keywords: Series-parallel structure, reliability, components independence, pairwise interchange, optimal/suboptimal components allocation.

1. Introduction

This paper studies the problem of optimal components allocation in a series-parallel reliability structure. Such a system is usually represented as a block diagram composed of serially arranged substructures, each of which consists of components arranged in parallel. All the components are assumed to be binary (either operable or failed) and independent (each component fails with a probability independent of the states of the remaining ones). The author presents a fast, nearly optimal method of allocating components with different failure probabilities to individual subsystems with the purpose of maximizing the system's reliability. Interestingly enough, while the optimal components allocation problem for a parallel-series system (i.e., composed of series subsystems arranged in parallel) has a well-known simple solution (see El Neweihi et al. (1986)), this is not the case with a series-parallel one.

The optimal allocation problem in reliability theory has been investigated by a number of researchers who proposed quite elaborate methods of its solution. Different approaches to optimizing not only series-parallel, but coherent systems in general, can be found in older papers, e.g. Boland et al. (1989), Bhattacharya and Samaniego (2008), Prasad and Raghavachari (1998), Lin and Kuo (2002), Yalaoui et al. (2005). Over the years, the considered problem has evolved into several variants. Many authors study the redundancy allocation problem for a series-parallel system (see Feizabadi and Jahromi (2017) and Zhang and Li (2022), where the problem definition and comprehensive literature surveys can be found). In a few papers, e.g. Fang et al. (2022), Zhang et al. (2022) or Lu et al. (2023), it is assumed that

the components are mutually dependent. Other authors, e.g. Yalaoui et al. (2005), assume that certain cost is involved in placing a component in a particular location in the reliability structure and attempt to find the allocation ensuring the minimal total cost. The issue of optimal components arrangement is also analyzed in the context of shock models. See Ling at al. (2021) for recent results and relevant bibliography.

In the current paper, it is assumed that independent components with given failure probabilities can be placed in arbitrary locations in a series-parallel structure. Two algorithms are presented that, starting from a random initial allocation and using pairwise interchanges of components between the parallel substructures, find the (sub)optimal allocation that (nearly) maximizes the system reliability. The difference between the optimal and suboptimal reliabilities can be estimated using the author's theorem that gives an upper bound on the optimal value. The operation of the algorithms is illustrated with several simple examples.

The optimization problem considered in this paper was earlier addressed in Kuo and Zuo (2003), where the system reliability was also improved by using pairwise interchanges, but no estimate of the difference between the optimal and suboptimal reliability was given, and the components interchange procedure proposed herein is likely more efficient.

2. Notation and the main theorem

The following notation will be used in the paper:

n – the number of parallel subsystems in the considered series-parallel system

m_j – the number of components in the j-th subsystem, j=1,…,n

q_{ij} – the failure probability of the i-th component in the j-th subsystem

Q_j – the set of failure probabilities of the components in the j-th subsystem, i.e. $Q_j=\{q_{ij}, i=1,\ldots,m_j\}$

Q – the set of failure probabilities of all the system's components, i.e. $Q=Q_1\cup\ldots\cup Q_n$

π_j – the product of all failure probabilities in Q_j, i.e. $\pi_j = \prod_{i=1,\ldots,m_j} q_{ij}$

π – the product of all failure probabilities in Q, i.e. $\pi = \prod_{j=1,\ldots,n} \pi_j$

R – the system reliability, i.e. $R = \prod_{j=1,\ldots,n}[1-\pi_j]$

We also set

$$Q_j^+ = Q\backslash(Q_1 \cup \ldots \cup Q_j),\ j = 1,\ldots,n$$

Theorem 1

Let the components of a series-parallel system have arbitrary failure probabilities such that their product is equal to π, where $0<\pi<1$, i.e.

$$\prod_{j=1}^{n}\prod_{i=1}^{m_j} q_{ij} = \pi \quad (1)$$

Then, the system reliability is maximized when the failure probability of each parallel subsystem is equal to $\pi^{1/n}$, i.e.

$$\prod_{i=1}^{m_j} q_{ij} = \pi^{1/n} \quad (2)$$

Hence, the maximum system reliability amounts to $[1-\pi^{1/n}]^n$.

Proof: The detailed proof will appear in the extended version of this paper, being prepared for publication.

Note that Eq. (2) holds if, for instance, the failure probability of each component in the j-th subsystem is equal to π to the power of $1/m_j{\cdot}n$. Also note that Theorem 1 assumes that the failure probabilities can take arbitrary values, provided that their product is equal to π, whereas the components of the considered system have fixed failure probabilities.

3. Two algorithms for optimizing the system reliability

It follows from Theorem 1 that the system reliability reaches its maximum if the components are arranged in such a way that the failure probabilities of the individual parallel subsystems are as close as possible to $\pi^{1/n}$. Such an arrangement can be obtained by composing the set Q_1 of those failure probabilities from Q that minimize $|\pi_1 - \pi^{1/n}|$, then composing the set Q_2 of those failure probabilities from $Q\setminus Q_1$ that minimize $|\pi_2 - \pi^{1/n}|$, etc. The final arrangement is thus a result of a (n–1)-step procedure, where Q_j is composed of those failure probabilities from $Q\setminus(Q_1\cup\ldots\cup Q_{j-1})$ that minimize $|\pi_j - \pi^{1/n}|$. We assume that $Q_1\cup\ldots\cup Q_{j-1}=\varnothing$ for j=1. The probabilities composing Q_j are selected by means of pairwise interchanges between Q_j and $Q\setminus(Q_1\cup\ldots\cup Q_{j-1})$. It should be noted that the obtained arrangement may be suboptimal rather

than optimal, because pairwise interchanges can lead to a local minimum of $|\pi_j - \pi^{1/n}|$, where the global one is over all possible arrangements.

We now present two algorithms implementing the above outlined procedure. They differ in the method of avoiding or leaving a local minimum of $|\pi_j - \pi^{1/n}|$.

Algorithm 1:

For j=1 to n–1 do {
 Initialize Q_j and π_j ; (see Remark 1)
 $Q_j^* \leftarrow Q_j$; $\pi_j^* \leftarrow \pi_j$;
 $k \leftarrow 1$;
 While *TRUE* do {
  Select any $q' \in Q_j$ and $q'' \in Q_j^+$ such that
  $\left|\pi_j \frac{q''}{q'} - \sqrt[n]{\pi}\right| < \left|\pi_j^* - \sqrt[n]{\pi}\right|$; (see Remark 2)
  If such q' and q'' exist
  then {
   $Q_j^* \leftarrow Q_j \setminus \{ q' \} \cup \{ q'' \}$;
   $\pi_j^* \leftarrow \pi_j \cdot q''/q'$;
   If j=n–1 and $\pi_1^* = \ldots = \pi_{n-1}^* = \pi^{1/n}$ then stop ;
  }
  else {
   $k \leftarrow k+1$;
   if $k > k\_max$ then break the while-loop ;
   Select any $q' \in Q_j$ and $q'' \in Q_j^+$;
  } (see Remark 3)
  $Q_j \leftarrow Q_j \setminus \{ q' \} \cup \{ q'' \}$;
  $Q_j^+ \leftarrow Q_j^+ \setminus \{ q'' \} \cup \{ q' \}$;
  $\pi_j \leftarrow \pi_j \cdot q''/q'$;
 } (end of while-loop)
} (end of for-loop)

Remark 1: Q_j is initialized with random values from $Q \setminus (Q_1^* \cup \ldots \cup Q_{j-1}^*)$

Remark 2: The search for q' and q'' satisfying this inequality stops when the first such pair is found.

Remark 3: If no pairwise interchange between Q_j and Q_j^+ can produce π_j closer to $\pi^{1/n}$ than π_j^*, then k is increased before the next cycle of the while-loop. The set Q_j obtained in the current cycle is the initial Q_j for the next cycle. Since the same Q_j may have been obtained in one of the previous cycles, the algorithm can fall into an infinite loop. For this reason, k is limited by k_max. Instead of increasing k and checking if it reached k_max, the alternative solution it to try to generate Q_j different from all the previous ones. However, this involves increasingly many backward comparisons, which affects the algorithm's efficiency. The possible workaround is to try to find Q_j such that the respective π_j is not repeated, since searching among previous values of π_j takes less time than searching among previous sets Q_j.

Algorithm 2:

For j=1 to n–1 do {
 Initialize Q_j and π_j ; (see Remark 1 to Alg. 1)
 $Q_j^* \leftarrow Q_j$; $\pi_j^* \leftarrow \pi_j$;
 $k \leftarrow 1$;
 While $k \le k\_max$ do {
  Select $q' \in Q_j$ and $q'' \in Q_j^+$ such that
  $(q', q'') = \underset{(q_1, q_2) \in Q_j \times Q_j^+}{\arg\min} \left|\pi_j \frac{q_2}{q_1} - \sqrt[n]{\pi}\right|$;
  (see Remark 1)
  Swap q' and q'', and update π_j, i.e.
  execute the following operations:
  $Q_j \leftarrow Q_j \setminus \{ q' \} \cup \{ q'' \}$;
  $Q_j^+ \leftarrow Q_j^+ \setminus \{ q'' \} \cup \{ q' \}$;
  $\pi_j \leftarrow \pi_j \cdot q''/q'$;
  If $\left|\pi_j - \sqrt[n]{\pi}\right| < \left|\pi_j^* - \sqrt[n]{\pi}\right|$ (see Remark 2)
  then {
   $Q_j^* \leftarrow Q_j$ (Q_j is the new candidate for Q_j^*) ;
   $\pi_j^* \leftarrow \pi_j$ (π_j is the new candidate for π_j^*) ;
   If j=n–1 and $\pi_1^* = \ldots = \pi_{n-1}^* = \pi^{1/n}$ then stop ;
  }
  else
   $k \leftarrow k+1$; (see Remark 3 to Alg. 1)
 } (end of while-loop)
} (end of for-loop)

Remark 1: Algorithm 2 can be called "greedy", because, unlike in Algorithm 1, each cycle of the while-loop begins with an attempt to find $q' \in Q_j$ and $q'' \in Q_j^+$ such that that $\pi_j \cdot q''/q'$ is as close as possible to $\pi^{1/n}$.

Remark 2: It can happen that π_j computed in the current cycle of the while-loop is farther from $\pi^{1/n}$ than π_j^* is.

Remark 3: Clearly, the minimum value of $|\pi_j \cdot q_2/q_1 - \pi^{1/n}|$ found in one cycle of the while-loop is local in the sense that it is a minimum over pairwise interchanges between Q_j and Q_j^+.

However, it is possible to get out of this local minimum in the subsequent cycles of the while-loop. Obviously, the greater the number of cycles the closer π_j^* can be to $\pi^{1/n}$.

Remark 4: Since Algorithm 1 choses any $q' \in Q_j$ and $q'' \in Q_j^+$ satisfying the inequality in the first command in the while-loop, it examines more different allocations than Algorithm 2. Thus, it may seem that the former is likely to find an allocation ensuring higher system reliability than the latter, although at the cost of more computing time. However, the examples presented in the next section and several others analyzed by the author suggest that the second algorithm performs no worse than the first, and finds the (sub)optimal solution in shorter time.

4. A few illustrating examples.

In all the examples *k_max* is set to 2.

Example 1
Let us consider a series-parallel system composed of two 4-element parallel subsystems, depicted in Fig. 1, which shows the initial allocation of the components' failure probabilities.

$$\begin{bmatrix}0.2\\0.2\\0.3\\0.3\end{bmatrix}\begin{bmatrix}0.4\\0.4\\0.7\\0.7\end{bmatrix}$$

Fig. 1. A system composed of two subsystems

It holds that $n=2$ and $\pi=(0.2\cdot0.3\cdot0.4\cdot0.7)^2$, hence $\pi^{1/n}=0.0168$. We first apply Algorithm 1 and present one of its possible realizations.

Cycle 1 of the for-loop:
$Q_1^*= Q_1=\{0.2, 0.2, 0.3, 0.3\}$
$\pi_1^*=\pi_1=0.0036$, $|\pi_1^* - \pi^{1/2}|=0.0132$
$k=1$
Cycle 1 of the while-loop:
$q'=0.2$, $q''=0.4$,
$Q_1^*=Q_1= \{0.4, 0.2, 0.3, 0.3\}$
$Q_1^+=\{0.2, 0.4, 0.7, 0.7\}$
$\pi_1^*=\pi_1=0.0072$, $|\pi_1^* - \pi^{1/2}|=0.0096$
Cycle 2 of the while-loop:
$q'=0.2$, $q''=0.4$
$Q_1^*= Q_1=\{0.4, 0.4, 0.3, 0.3\}$
$Q_1^+=\{0.2, 0.2, 0.7, 0.7\}$
$\pi_1^*=\pi_1=0.0144$, $|\pi_1^* - \pi^{1/2}|=0.0024$
Cycle 3 of the while-loop:
π_1^* cannot be improved
$q'=0.4$, $q''=0.7$
$k=2$
$Q_1=\{0.7, 0.4, 0.3, 0.3\}$
$Q_1^+=\{0.2, 0.2, 0.4, 0.7\}$
$\pi_1=0.0252$
Cycle 4 of the while-loop:
$q'=0.3$, $q''=0.2$
$Q_1^*=Q_1=\{0.7, 0.4, 0.2, 0.3\}$
$Q_1^+=\{0.3, 0.2, 0.4, 0.7\}$
$\pi_1^*=\pi_1=0.0168$, $|\pi_1^* - \pi^{1/2}|=0$

Since $n=2$, the for-loop ends after one cycle. The value of π_1^* computed in cycle 4 of the while-loop is equal to $\pi^{1/2}$. The algorithm stops at this point, because, according to Theorem 1, $(1-\pi^{1/2})^2 = 0.966682$ is the highest possible system reliability. For comparison, the initial allocation yields the reliability equal to 0.918282. Let us note that it is impossible to improve π_1^* in cycle 3, because if $Q_1=\{0.4, 0.4, 0.3, 0.3\}$ then no pairwise interchange between Q_1 and Q_1^+ causes $\pi_1 \cdot q''/q'$ to be closer to $\pi^{1/n}$ than π_1.

Example 2
The system and its initial configuration are the same as in Example 1. We now apply Algorithm 2 that operates as follows:

Cycle 1 of the for-loop:
$Q_1^*= Q_1=\{0.2, 0.2, 0.3, 0.3\}$,
$\pi_1^*=\pi_1=0.0036$, $|\pi_1^* - \pi^{1/2}|=0.0132$
$k=1$
Cycle 1 of the while-loop:
$q'=0.2$, $q''=0.7$
$Q_1^*=Q_1= \{0.7, 0.2, 0.3, 0.3\}$
$Q_1^+=\{0.4, 0.4, 0.2, 0.7\}$
$\pi_1^*=\pi_1=0.0126$, $|\pi_1^* - \pi^{1/2}|=0.0042$
Cycle 2 of the while-loop:
$q'=0.3$, $q''=0.4$
$Q_1^*= Q_1=\{0.7, 0.2, 0.4, 0.3\}$
$Q_1^+=\{0.3, 0.4, 0.2, 0.7\}$
$\pi_1^*=\pi_1=0.0168$, $|\pi_1^* - \pi^{1/2}|=0$

Now the optimal allocation was found in cycle 2 of the while-loop. It is easy to check that if $Q_1=\{0.4, 0.4, 0.3, 0.3\}$ is the initial allocation for

Algorithm 2, then cycles 1 and 2 of the while-loop yield the same results as cycles 3 and 4 in Example 1 (as regards cycle 3, see Remark 2 to Algorithm 2).

Example 3
We now consider a system whose structure and initial allocation of the failure probabilities are illustrated in Fig. 2.

$$\begin{bmatrix} 0.2 \\ 0.25 \end{bmatrix} \begin{bmatrix} 0.3 \\ 0.4 \\ 0.5 \end{bmatrix} \begin{bmatrix} 0.6 \\ 0.7 \\ 0.8 \end{bmatrix}$$

Fig. 2. A system composed of three subsystems

It holds that n=3 and $\pi^{1/n}$=0.100266. One of possible realization of Algorithm 1 is presented below.

Cycle 1 of the for-loop:
Q_1*= Q_1={0.2, 0.25}
π_1*=π_1=0.05, $|\pi_1$* – $\pi^{1/3}|$=0.050266
k=1
Cycle 1 of the while-loop:
q'=0.25, q''=0.4,
Q_1*=Q_1={0.2, 0.4}
Q_1^+={0.25, 0.3, 0.5, 0.6, 0.7, 0.8}
π_1*=π_1=0.08, $|\pi_1$* – $\pi^{1/2}|$=0.020266
Cycle 2 of the while-loop:
q'=0.4, q''=0.5,
Q_1*=Q_1={0.2, 0.5}
Q_1^+={0.25, 0.3, 0.4, 0.6, 0.7, 0.8}
π_1*=π_1=0.1, $|\pi_1$* – $\pi^{1/2}|$=0.000266
Cycle 3 of the while-loop:
π_1* cannot be improved
k=2
q'=0.5, q''=0.6
Q_1={0.2, 0.6}
Q_1^+={0.25, 0.3, 0.4, 0.5, 0.7, 0.8}
π_1=0.12
Cycle 4 of the while-loop:
π_1* cannot be improved
k=3
break the while-loop

Cycle 2 of the for-loop:
Q_2*=Q_2={0.25, 0.3, 0.4}
π_2*=π_2=0.03, $|\pi_2$* – $\pi^{1/3}|$=0.070266
k=1
Cycle 1 of the while-loop:
q'=0.25, q''=0.7,
Q_2*=Q_1={0.7, 0.3, 0.4}
Q_2^+={0.25, 0.6, 0.8}
π_2*=π_2=0.084, $|\pi_2$* – $\pi^{1/2}|$=0.016266
Cycle 2 of the while-loop:
q'=0.7, q''=0.8,
Q_2*=Q_2={0.8, 0.3, 0.4}
Q_2^+={0.25, 0.6, 0.7}
π_2*=π_2=0.096, $|\pi_2$* – $\pi^{1/2}|$=0.004266

Cycle 3 of the while-loop:
π_2* cannot be improved
k=2
q'=0.3, q''=0.25
Q_2= {0.8, 0.25, 0.4}
Q_2^+={0.3, 0.6, 0.7}
π_2=0.08
Cycle 4 of the while-loop:
π_2* cannot be improved
k=3
break the while-loop

Thus, Q_1={0.2, 0.5}, Q_2={0.8, 0.3, 0.4}, Q_3={0.25, 0.6, 0.7} is the final allocation found by Algorithm 1. It yields the system reliability equal to 0.728172. It can be checked that this is one of two optimal allocations, where the second is Q_1={0.25, 0.4}, Q_2={0.3, 0.7, 0.5}, Q_3={0.2, 0.6, 0.8}. For comparison, the initial allocation yields the reliability equal to 0.592952.

Example 4
Now, the subsequent steps of Algorithm 2 applied to the system in Fig. 2 will be demonstrated.

Cycle 1 of the for-loop:
Q_1*= Q_1={0.2, 0.25}
π_1*=π_1=0.05, $|\pi_1$* – $\pi^{1/3}|$=0.050266
k=1
Cycle 1 of the while-loop:
q'=0.2, q''=0.4,
Q_1*=Q_1={0.4, 0.25}
Q_1^+={0.2, 0.3, 0.5, 0.6, 0.7, 0.8}
π_1*=π_1=0.1, $|\pi_1$* – $\pi^{1/3}|$=0.000266
Cycle 2 of the while-loop:
q'=0.25, q''=0.3,
Q_1={0.4, 0.3}
Q_1^+={0.2, 0.25, 0.5, 0.6, 0.7, 0.8}
π_1=0.12, $|\pi_1 - \pi^{1/3}|$=0.019734
π_1* cannot be improved
k=2

Cycle 3 of the while-loop:
$q'=0.3$, $q''=0.25$
$Q_1=\{0.4, 0.25\}$
$Q_1^+=\{0.2, 0.3, 0.5, 0.6, 0.7, 0.8\}$
$\pi_1=0.1$, $|\pi_1 - \pi^{1/3}|=0.000266$
π_1* cannot be improved
$k=3$
stop the while-loop

Cycle 2 of the for-loop:
Q_2*= $Q_2=\{0.2, 0.3, 0.5\}$
π_2*=$\pi_2=0.03$, $|\pi_2$* $- \pi^{1/3}|=0.070266$
$k=1$
Cycle 1 of the while-loop:
$q'=0.2$, $q''=0.7$,
Q_2*=$Q_2=\{0.7, 0.3, 0.5\}$
$Q_2^+=\{0.2, 0.6, 0.8\}$
π_2*=$\pi_2=0.105$, $|\pi_2$* $- \pi^{1/3}|=0.004734$
Cycle 2 of the while-loop:
$q'=0.7$, $q''=0.8$,
$Q_2=\{0.8, 0.3, 0.5\}$
$Q_2^+=\{0.2, 0.6, 0.7\}$
$\pi_2=0.12$, $|\pi_2 - \pi^{1/3}|=0.019734$
π_2* cannot be improved
$k=2$
Cycle 3 of the while-loop:
$q'=0.8$, $q''=0.7$
$Q_2=\{0.7, 0.3, 0.5\}$
$Q_2^+=\{0.2, 0.6, 0.8\}$
$\pi_2=0.105$, $|\pi_2 - \pi^{1/3}|=0.004734$
π_2* cannot be improved
$k=3$
stop the while-loop

Thus, $Q_1=\{0.4, 0.25\}$, $Q_2=\{0.7, 0.3, 0.5\}$, $Q_3=\{0.2, 0.6, 0.8\}$ is the final allocation found by Algorithm 2. It yields the same system reliability as the allocation found by Algorithm 1. Let us note that the allocations found in cycles 1 and 3 of the while-loop in both iterations of the for-loop are identical. In consequence, without a limit on the variable k, Algorithm 2 would fall into an infinite loop.

5. Concluding remarks

Two algorithms have been presented that find a suboptimal or optimal arrangement of components in a series-parallel system. The components are assumed to be independent and each component can be placed in any parallel subsystem. The operation of both algorithms is analyzed in detail for two small systems. Although in all four cases it took no more than a few steps to find optimal allocations, it can happen that only a suboptimal one will be found for a more complex system. However, sufficiently large *k_max* and appropriate technique of infinite loop avoidance can be a way to reach an optimal solution. It can be concluded from the provided examples that Algorithm 2 is more efficient than Algorithm 1, although this should be confirmed by more tests with larger systems. It seems difficult, if possible at all, to compare the algorithms' efficiency by mathematical reasoning. The presented method, after appropriate modifications, seems to be applicable to different variants of optimal allocation problem, mentioned in the Introduction. This will be the topic of further research.

References

Bhattacharya, D. and F.J. Samaniego. (2008). "On the optimal allocation of components within coherent systems." *Statistics and Probability Letters* 78, 938–943.

Boland, P.J., F. Proschan, and Y.L. Tong. (1989). "Optimal Arrangement of Component via Pairwise Rearrangements." *Naval Research Logistics* 36, 807-815.

El-Neweihi, E., F. Proschan, and J. Sethuraman. (1986). "Optimal allocation of components in parallel–series and series–parallel systems." *Journal of Applied Probability* 23, 770-777.

Fang, L., X. Zhang, and Q. Jin. (2022). "Optimal Grouping of Heterogeneous Components in Series and Parallel Systems Under Archimedean Copula Dependence." *Joural of Systems Science and Complexity* 35, 1030-1051.

Feizabadi, M. and A.E. Jahromi. (2017). "A new model for reliability optimization of series-parallel systems with non-homogeneous components." *Reliability Engineering and System Safety* 157, 101-112.

Kuo, W. and M.J. Zuo. (2003). *Optimal Reliability Modeling. Principles and Applications*. John Wiley & Sons.

Lin, F.-H. and W. Kuo. (2002). "Reliability Importance and Invariant Optimal Allocation." *Journal of Heuristics* 8, 155-171.

Ling, X., Y. Zhang, and Y. Gao. (2021). "Reliability optimization in series-parallel and parallel-series systems subject to random shocks." *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 235, 998-1008.

Lu, B., J. Zhang, and R. Yang (2023). "Optimal allocation of a coherent system with statistical dependent subsystems." *Probability in the*

Engineering and Informational Sciences 37, 29-48.

Prasad, V.R. and M. Raghavachari. (1998). "Optimal Allocation of Interchangeable Components in a Series-Parallel System." *IEEE Transactions on Reliability* 47, 255-260.

Zhang, H. and Y.-F. Li, (2022). "Approximate method for redundancy allocation problem in multi-state series-parallel system." *Safety and Reliability of Systems and Processes. 16th Summer Safety & Reliability Seminars - SSARS 2022*, 215-224.

Zhang, J., R. Yan, and J. Wang. (2022). "Reliability optimization of parallel-series and series-parallel systems with statistically dependent components." *Applied Mathematical Modelling* 102, 618-639.

Yalaoui, A., C. Chu, and E. Chatelet. (2005). "Reliability allocation problem in a series–parallel system." *Reliability Engineering and System Safety* 90, 55-61.