Proceedings of the ASME 2018 13th International Manufacturing Science and Engineering Conference
MSEC2018
June 18-22, 2018, College Station, TX, USA

MSEC2018-6481

Redundancy Allocation for Serial-parallel System Considering Heterogeneity of Components

Zhicheng Zhu
Department of Industrial Engineering
Lamar University
Beaumont, TX, 77710, USA

Yisha Xiang
Department of Industrial Engineering
Lamar University
Beaumont, TX, 77710, USA

David Coit
Department of Industrial and Systems Engineering
Rutgers University
Piscataway, NJ, 08844, USA

Corresponding author: Yisha Xiang, Email: yxiang@lamar.edu, Phone: 409-880-8475

ABSTRACT

The redundancy allocation problem (RAP) for series-parallel system is a system design problem by selecting an appropriate number of components from multiple choices for desired objectives, such as maximizing system reliability, minimizing system cost. RAP has been extensively studied in the last decades. The majority of existing RAP models assume that components for selection are from homogeneous populations. However, due to manufacturing difficulties and variations in raw materials, many manufactured components/parts are heterogeneous, consisting of multiple subpopulations. In this research, we consider a typical RAP with the objective of maximizing the system reliability subject to the constraint of system cost. Components in each choice are assumed to be degradation-based, and each choice consists one normal subpopulation and several abnormal subpopulations. Numerical examples are investigated to illustrate the impact of the component heterogeneity.

KEY WORDS: Redundancy allocation problem, heterogeneity, Stochastic Degradation

1. INTRODUCTION

In the redundancy allocation problem (RAP), a system contains a certain number of subsystems in series. Each subsystem has multiple choices for selection, and components within a subsystem are connected in parallel. Choices are usually characterized by reliability, cost and weight, etc. Therefore, RAP for series-parallel system is a system design problem of selecting an appropriate number of components from multiple choices in order to achieve a desired goal, such as maximization of system reliability, and minimization of system cost. In this research, we consider components from heterogeneous populations [1], and aim to maximize system reliability with a budget constraint.

The formulation of RAP aims to maximize a desired objective but subject to some constraints. However, the component reliability is often given as a priori, and its dynamic behavior is often ignored in most existing RAP models. The RAP problem formulation and assumption vary from one research to another.

There are two types of original system topology. One assumes that there are h_i components that are already in subsystem i [2]. The other considers zero components originally, but at least k_i components must be selected for subsystem i [3]. The state of a component can be binary, which is either good or failed. Typically, components are characterized by their reliability to represent the good or failed state [4]. RAP considering multi-state components is also widely studied [5]. Hierarchical performance level is employed to evaluate how good the components and system are. Generally, the objective in multi-state RAP is to minimize the total cost while insuring system performs above a certain level [6]. Components in a subsystem could be restricted from one choice or allowed to be selected from different choices, i.e., mixing components. Fyffe *et al.* [7] give the optimal solution by applying Lagrangian multipliers if using the same components in a subsystem, with constraints on weight and cost. Coit *et al.* [4] show that a better system reliability can be achieved if mixing components is allowed. Mixing components results in a larger solution space, which is difficult to solve. Comparing to the analytical approach, which needs to make approximations on objective function, heuristic algorithm, i.e. genetic algorithm (GA), yields a reasonably good result in this problem [8].

However, none of these afore-mentioned models considers how each component reliability is evaluated. In addition, the majority of existing RAP models assume that components for selection are from homogeneous populations. However, due to manufacturing difficulties and variations in raw materials, many manufactured components/parts are heterogeneous, consisting of several subpopulations: early failures where the mean time to failure (MTTF) is relatively 'short' and wear-out failures where the MTTF is relatively 'long'. For example, certain classes of semiconductor devices tend either to fail very early or to last a relatively long time. Many Micro-Electro-Mechanical Systems (MEMS) devices have also been found to have two subpopulations [9-11] due to the difficulty in manufacturing processes and variability in raw materials. Figure 1 illustrates the degradation paths from two subpopulations.

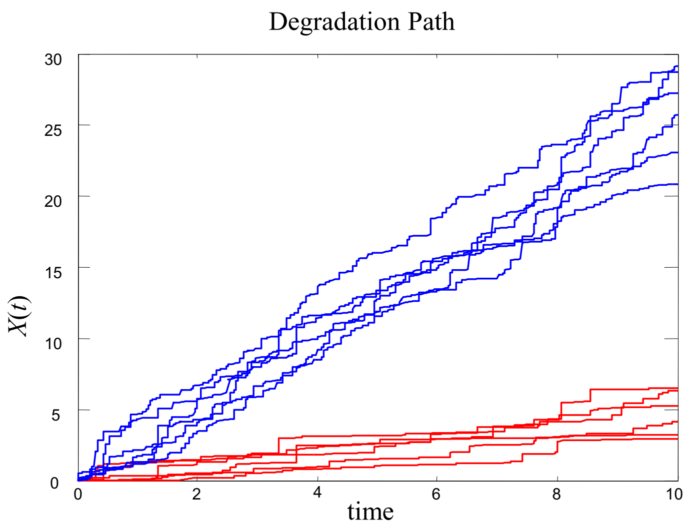

Figure 1: Degradation paths with subpopulations

In this research, we consider a typical RAP whose objective is to maximize the system reliability subject to the constraint of system cost. Components in each choice are assumed to be degradation-based, and each choice consists of one normal subpopulation and several abnormal subpopulations. We investigate the impacts of component heterogeneity by comparing the system reliabilities from cases where the heterogeneity is considered and cases where the heterogeneity is ignored.

The remainder of the paper is organized as follows. Section 2 formulates the RAP model with consideration of components' heterogeneity. Section 3 presents the genetic algorithm used to seek the optimal system configuration. Numerical examples are provided in Section 4. Conclusions are given in Section 5.

2. MODEL DEVELOPMENT

As components deteriorate, a system becomes less reliable. It's commonly seen in lots of manufacturing machines. Adding redundant components is an efficient approach to improve system reliability. Therefore, we consider a combined series and parallel system. The system consists of s subsystems in series, and each subsystem has h_i identical components in parallel originally in subsystem i.

To improve the system reliability, we can add some number of components to each subsystem within the budget allowed. Let m_i denote the maximum choices for subsystem i, x_{ij} denote the number of components added to subsystem i from choice j. For each subsystem, all components stay in parallel and each subsystem is a 1-out-of-n: G system, which means at least one component functioning can ensure subsystem in operation. All redundancy is in hot standby with the component's deterioration rate remaining the same regardless whether in use or not.

As mentioned before, most researchers ignore the heterogeneity of components. Therefore, we make following reasonable assumptions that are commonly seen reliability problems.

1. All components are degradation-based, and their deterioration processes can be described by some stochastic processes. Although how the reliability of each component is calculated does not matter from the perspective of problem solving, evaluation of component's reliability is a big problem in practice.
2. For subsystem i, there are n_{ij} subpopulations in choice j among m_i choices. All the choices have one normal subpopulation and several abnormal subpopulations. If the engineer is not aware of the existence of abnormal subpopulations, the reliability of components will be calculated only based on the normal subpopulation.
3. Failed components do not damage the system and are not repairable.
4. Mixture of components is allowed for a subsystem. For each subsystem, components can come from different choices, which can potentially achieve a higher reliability compared to the condition of allowing only one choice.
5. The cost of components from choice j for subsystem i is c_{ij}. A choice with a slower degradation rate is more expensive.

In most industries, what we care about is how to maintain the system functioning for a specific time period. For example, in aerospace industry, we need make sure no component goes wrong during the launching period for the launching system. The system required mission time is noted as T_0. The whole problem, noted as P1, can be formulated as below:

P1: Maximize the mission success probability

$$\max \Pr\{t > T_o\} = R\left(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_s, T_0\right)$$

Decision Variable: x_{ij}

$$\text{s.t.} \sum_{i=1}^{s}\sum_{j=1}^{m_i} x_{ij}c_{ij} \le C$$

$R\left(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_s, T_0\right)$ is the reliability function of the system, and $\mathbf{x}_i = \left(x_{i1}, x_{i2}, \ldots, x_{im_i}\right)$ represents the vector of components' number from each choice. The problem is to find out all x_{ij}, which are the component numbers from choice j in subsystem i. The constraint ensures that the total cost of components cannot exceed the budget C.

Because subsystems are in series, we have,

$$R\left(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_s, T_0\right) = \prod_{i=1}^{s} R_i\left(\mathbf{x}_i, T_0\right),$$

where $R_i\left(\mathbf{x}_i, T_0\right)$ is the reliability function for subsystem i. In subsystem i, there are h_i identical components originally and x_i new components. Therefore, we obtain

$$R_i\left(\mathbf{x}_i, T_0\right) = 1 - \left(1 - R_{h_i}\left(T_0\right)\right)^{h_i} \prod_{j=1}^{m_i}\left(1 - R_{ij}\left(T_0\right)\right)^{x_{ij}}$$

,

where $R_{h_i}\left(T_0\right)$ is the reliability function of original subsystem i, and $R_{ij}\left(T_0\right)$ is the reliability function of new components from choice i in subsystem j.

There are n_{ij} subpopulations in choice j for subsystem i, then

$$R_{ij}\left(T_0\right) = \sum_{k=1}^{n_{ij}} p_{ijk} R_{ijk}\left(T_0\right).$$

where p_{ijk} is the proportion of subpopulation k in choice j for subsystem i.

3. Genetic Algorithm

In this paper, we use GA to seek the optimal solutions. In GA, we use fitness function to simulate the natural selection and operation, e.g., crossover to simulate the reproduction process. After a long time, these genes will be the majority in the species population. Moreover, random changes in genes referred to as gene mutation takes place in nature. A good mutation provides a better chance in surviving, while an unfit mutation will surely be filtered by natural selection. In GA, every gene in child has a probability to mutate, known as mutation rate. The process is referred to as mutation.

In our encoding, we need represent every component in vector v_q since mixed components are allowed in each subsystem. In every component's position, choice number is the coded value, which ranges from one to m_i for subsystem i. Zero is used to represent empty position. And n_{max} is predetermined to define the maximum component number for all subsystem.

For example, $s = 3$, $m_1 = 4$, $m_2 = 2$, $m_3 = 3$, $n_{max} = 5$, v_q = (3 3 2 2 1|2 1 0 0 0| 2 2 2 0 0) represent a system that has 3 subsystems. For subsystem 1, subsystem 2, subsystem 3, there are 4, 2, 3 choices, respectively. In subsystem 1, there is 1 component from choice 1, 2 components from choice 2 and 2 components from choice 3. In subsystem 2, there is 1 component from choice 1, and 1 component from choice 2. In subsystem 3, there are 3 components from choice 2. All zeros represent no component in that position. Details of each step are provided in the next section.

The genetic algorithm has the following four steps.

Step 1: Generate initial population
Step 2: Crossover
Step 3: Mutation
Step 4: Cull infeasible child
Step 5: Generate new population

3.1 Generate initial population

Randomly generate multiple solutions v_q based on n_{max} and population size P. The whole population contains P solutions. In each solution, subsystem has no more than n_{max} components. For example, when s=3, m_1=4, m_2=2, m_3=3, we have n_{max} = 5. For subsystem 1, we generate n_{max} = 5 random integers range in [0, 4], i.e. [1 0 3 4 2]. Then, we sort this vector in descending order, which is [4 3 2 1 0]. Therefore, we can get the choices in subsystem 1 and 2. Then we calculate the reliability and cost in each solution.

3.2 Crossover

To perform the crossover, we first sort current population in ascending order based on their reliability. We then randomly generate two different values U_1 and U_2 in range 1 to P as the index of parents. Each subsystem of the child is randomly selected from parents. Only one child is generated in each crossover operation. For example, s=3, m_1=4, m_2=2, m_3=3, we have n_{max} = 5.

Parent 1: v_q = (**3 3 2 2 1**|2 1 0 0 0| **2 2 2 0 0**)
Parent 2: v_q = (4 2 0 0 0|**2 2 2 1 0**| 3 3 1 0 0)
Child: v_q = (3 3 2 2 1|2 2 2 1 0| 2 2 2 0 0)

3.3 Mutation

Each value in the values of a child has a probability, which is called the mutation rate, to change to other possible choices excluding itself, e.g., v_q = (3 **3** 2 2 1|2 1 0 0 0| 2 2 2 0 0). Suppose that the second component in subsystem 1 is mutating. Then it has 3 choices to mutate with equal probability, which are 0, 1, 2 and 4 if the maximum number of choices is 4. If it mutates into 0, v_q = (3 **0** 2 2 1|2 1 0 0 0| 2 2 2 0 0), then sort each subsystem in v_q in descending order, which is v_q = (3 2 2 1 **0**|2 1 0 0 0| 2 2 2 0 0).

3.4 Cull infeasible child

Check the feasibility of the child. Only if it is feasible, we add it to the offspring. If the size of offspring is less than $2P$, keep doing crossover.

3.5 Generate new population

From previous steps, we obtain P solutions, which have the highest reliability from offspring to generate a new population. If current population generations exceed the maximum generation number, stop. Otherwise, go to step 2.

4 NUMERICAL EXAMPLES

To assess the impacts of component heterogeneity on the RAP, we conduct the analysis through some numerical examples.

4.1 Example 1:

Consider a system that contains 3 subsystems, which have 2, 4, 3 component choices. Assume that component degradation can be modeled by a gamma process, which is characterized by shape parameter α and scale parameter β. The mission time T_0 is 50 and system budget C is 50.

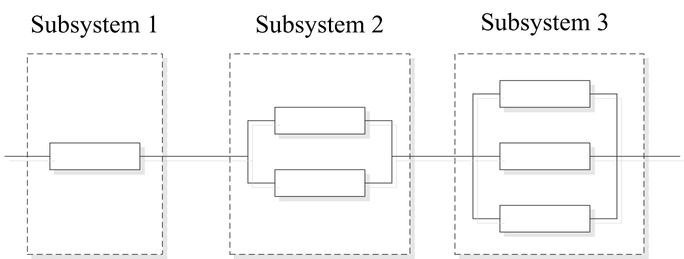

Figure 2: System initial configuration in examples

The subsystem information is shown in Table 1 and the initial configuration of system is illustrated in Figure 2. Take subsystem 1 as an example. Subsystem 1 has one component originally, which is noted as h. The shape parameter α and scale parameter β are 1 and 2 respectively with a failure threshold D equals to 50. There are two choices for this subsystem. Both of them have two subpopulations with the same β, D and subpopulation proportion p but a different α. For the convenience of comparison, we use subpopulation 1 in all choices as the normal population and let $\alpha = 1$ in choice 1 in subsystem 1, and subpopulation 2 as the abnormal subpopulation and let $\alpha = 1.5$ in choice 1 in subsystem 1. A lower α in the normal subpopulation has a higher cost. For subsystem 1, it costs 9 in each component from choice 1 and 5 from choice 2, since α in subpopulation 1 in choice 1 is smaller than choice 2.

Table 1: System Parameters

subsystem 1							
		quantity	α	β	D	p	cost
h		1	1	2	50	NA	NA
choice	1	NA	[1, 1.5]	[2,2]	[50,50]	[0.9,0.1]	9
	2		[1.5, 2]	[2,2]	[50,50]	[0.9,0.1]	5
subsystem 2							
		quantity	α	β	D	p	cost
h		2	1	2	50	NA	NA
choice	1	NA	[1.1,1.6]	[2,2]	[50,50]	[0.9,0.1]	8
	2		[1.3,1.8]	[2,2]	[50,50]	[0.9,0.1]	6
	3		[1.5,2]	[2,2]	[50,50]	[0.9,0.1]	4
	4		[1.7,2.2]	[2,2]	[50,50]	[0.9,0.1]	2
subsystem 3							
		quantity	α	β	D	p	cost
h		3	1	2	50	NA	NA
choice	1	NA	[1.2,1.7]	[2,2]	[50,50]	[0.9,0.1]	7
	2		[1.4,1.9]	[2,2]	[50,50]	[0.9,0.1]	5
	3		[1.6,2.1]	[2,2]	[50,50]	[0.9,0.1]	3

In GA, the population size is 40, and the offspring size is 80. Mutation rate is 0.4. Maximum population generations are 40 and n_{max} is 5.

We give an example where engineers do not know the heterogeneity situation in each choice, we refer to it as scenario 1. The engineers consider that there is only one subpopulation, which is the normal population in each choice. Therefore, engineers are led to wrong reliability. Ten replicate results are shown in Table 2 for 10 GA replicates. Column of reliability is the optimal solution in this replicate, with subsystem topology in column solution and system cost. We also show the actual system reliability in the third column. Averagely, the actual system reliability is 99.9% of the given optimal reliability.

Table 2: Result of Scenario 1 with a proportion [0.9, 0.1]

	reliability	actual reliability	cost	solution(subsystem)		
				1	2	3
1	0.9946	0.9915	49	2 1 1 0 0	4 3 2 2 1	0 0 0 0 0
2	0.9909	0.9892	49	1 1 1 0 0	4 4 3 2 0	3 2 0 0 0
3	0.9947	0.9907	49	1 1 0 0 0	2 1 0 0 0	2 2 1 0 0
4	0.9949	0.9912	47	1 1 0 0 0	4 4 1 1 0	3 3 3 0 0
5	0.9924	0.9886	48	1 1 0 0 0	4 3 3 2 2	3 3 3 0 0
6	0.9937	0.9896	49	1 1 0 0 0	4 3 1 0 0	2 2 1 0 0
7	0.9949	0.9914	49	1 1 0 0 0	3 3 3 1 1	3 0 0 0 0
8	0.994	0.9906	49	2 1 1 0 0	4 4 4 2 1	3 3 0 0 0
9	0.9954	0.9916	48	1 1 0 0 0	2 2 1 0 0	3 1 0 0 0
10	0.9947	0.9907	50	1 1 0 0 0	4 4 2 1 0	3 3 1 0 0
Average	0.99402	0.99051				
Max	0.9954	0.9916				
Min	0.9909	0.9886				
Variance	1.72E-06	9.71E-07				

Table 3 shows the results when engineers know the heterogeneity information in each choice, we refer to it as scenario 2. Given the same parameters, the average reliability in ten replicates is 0.99116, which is higher than the actual reliability 0.99051 in scenario 1.

Table 3: Result of Scenario 2 with proportion [0.9, 0.1]

	reliability	cost	solution(subsystem)		
			1	2	3
1	0.9914	50	2 1 1 0 0	3 3 2 1 0	2 0 0 0 0
2	0.9908	49	2 1 1 0 0	4 3 2 1 0	3 3 0 0 0
3	0.9921	50	1 1 0 0 0	4 4 4 1 1	3 1 0 0 0
4	0.992	50	1 1 0 0 0	4 2 1 1 0	3 2 0 0 0
5	0.9925	49	1 1 1 0 0	4 4 4 1 0	3 2 0 0 0
6	0.9899	49	1 1 1 0 0	4 3 3 2 0	3 3 0 0 0
7	0.9889	47	1 1 0 0 0	4 1 0 0 0	2 1 1 0 0
8	0.9906	49	1 1 0 0 0	3 2 1 0 0	3 2 2 0 0
9	0.9916	48	1 1 0 0 0	4 2 1 1 0	3 3 0 0 0
10	0.9918	50	1 1 0 0 0	3 3 1 1 0	3 2 0 0 0
Average	0.99116				
Max	0.9925				
Min	0.9889				
Variance	1.12E-06				

We can conclude that the consideration of reliability calculation is important, especially when heterogeneous subpopulations exist in components. Potential improvement of reliability in system reliability can be achieved when we fully consider the heterogeneity of components.

4.2 Example 2:

In the second numerical example, all parameters remain the same as in numerical example 1, except the proportion of subpopulation. The proportions of subpopulations 1 and 2 change from [0.9, 0.1] into [0.6, 0.4].

Table 4 shows the result of scenario 1 when proportion of subpopulation is [0.6, 0.4]. The actual reliability is only 97.8% of the optimal reliability. In comparison with the percentage in Table 2, where the actual reliability is 99.9% of the optimal reliability, the increased proportion of abnormal subpopulation yields a larger deviation from the actual reliability.

Table 4: Result of Scenario 1 with proportion [0.6, 0.4]

	reliability	actual reliability	cost	solution(subsystem)		
				1	2	3
1	0.9953	0.9749	50	2 1 1 0 0	4 2 1 1 0	3 0 0 0 0
2	0.9956	0.9723	49	1 1 0 0 0	3 2 2 1 0	1 0 0 0 0
3	0.9938	0.9713	47	2 1 1 0 0	4 3 1 0 0	3 1 0 0 0
4	0.9961	0.9727	50	1 1 0 0 0	3 1 1 0 0	2 1 0 0 0
5	0.9957	0.9721	49	1 1 0 0 0	1 1 0 0 0	2 2 2 0 0
6	0.9951	0.9739	50	2 1 1 0 0	4 2 2 1 0	2 0 0 0 0
7	0.9937	0.9692	50	1 1 0 0 0	4 3 1 0 0	3 3 2 1 0
8	0.9947	0.9819	48	1 1 1 0 0	4 3 3 1 0	3 0 0 0 0
9	0.9955	0.9741	49	2 1 1 0 0	4 1 1 0 0	3 2 0 0 0
10	0.9948	0.9726	46	2 1 1 0 0	4 2 1 0 0	1 0 0 0 0
Average	0.99503	0.9735				
Max	0.9961	0.9819				
Min	0.9937	0.9692				
Variance	5.66E-07	1.01E-05				

Table 5 shows the result of scenario 2 when subpopulation proportion is [0.6, 0.4]. Comparing with the actual reliability in Table 4, the optimal reliability when knowing the heterogeneity information can result in a better system reliability. Numerical example 2 further confirms the necessity of considering components' heterogeneity.

Table 5: Result of Scenario 2 with proportion [0.6, 0.4]

	reliability	cost	solution(subsystem)		
			1	2	3
1	0.9814	49	1 1 1 0 0	3 1 0 0 0	2 2 0 0 0
2	0.9775	50	1 1 1 0 0	4 4 2 0 0	3 2 2 0 0
3	0.9808	50	1 1 1 0 0	4 4 1 0 0	3 3 2 0 0
4	0.972	48	2 1 1 0 0	4 3 3 1 0	1 0 0 0 0
5	0.9804	49	1 1 1 0 0	2 2 0 0 0	3 1 0 0 0
6	0.9814	50	2 1 1 1 0	4 1 0 0 0	3 2 0 0 0
7	0.9785	49	2 1 1 1 0	4 4 2 0 0	1 0 0 0 0
8	0.9788	48	1 1 1 0 0	4 4 3 2 0	1 0 0 0 0
9	0.9708	50	2 1 1 0 0	4 1 0 0 0	3 1 1 0 0
10	0.9721	47	1 1 0 0 0	1 1 0 0 0	3 3 1 0 0
Average	0.97737				
Max	0.9814				
Min	0.9708				
Variance	1.57E-05				

5. CONCLUSION

In this research, we consider a typical RAP whose objective is to maximize the system reliability subject to the constraint of system cost. Components in each choice are assumed to be degradation based, and each choice consists one normal subpopulation and several abnormal subpopulations. Numerical analysis is conducted to assess the necessity of considering components' heterogeneity. Our numerical examples show that ignoring the heterogeneity in components can lead to inferior system reliability.

In this research, we consider a simple parallel system for illustration purpose. In future, more numerical experiments can be conducted to assess the impacts of components' heterogeneity on different system structures. It is also worth considering a joint optimization of redundancy allocation and preventive maintenance with the presence of components' heterogeneity.

ACKNOWLEDGEMENTS

This study is partially supported by the U.S. National Science Foundation through Award 1728257.

REFERENCES

1. Xiang, Y., D.W. Coit, and Q. Feng, *n Subpopulations experiencing stochastic degradation: reliability modeling, burn-in, and preventive replacement optimization.* IIE Transactions, 2013. **45**(4): p. 391-408.
2. Levitin, G., *Multistate series-parallel system expansion-scheduling subject to availability constraints.* IEEE Transactions on Reliability, 2000. **49**(1): p. 71-79.
3. Coit, D.W. and A.E. Smith, *Reliability optimization of series-parallel systems using a genetic algorithm.* IEEE Transactions on reliability, 1996. **45**(2): p. 254-260.
4. Coit, D.W. and A.E. Smith, *Redundancy allocation to maximize a lower percentile of the system time-to-failure distribution.* IEEE Transactions on Reliability, 1998. **47**(1): p. 79-87.
5. Levitin, G., et al., *Redundancy optimization for series-parallel multi-state systems.* IEEE Transactions on Reliability, 1998. **47**(2): p. 165-172.
6. Lisnianski, A. and Y. Ding, *Redundancy analysis for repairable multi-state system by using combined stochastic processes methods and universal generating function technique.* Reliability Engineering & System Safety, 2009. **94**(11): p. 1788-1795.
7. Fyffe, D.E., W.W. Hines, and N.K. Lee, *System reliability allocation and a computational algorithm.* IEEE Transactions on Reliability, 1968. **17**(2): p. 64-69.
8. Zia, L. and D.W. Coit, *Redundancy allocation for series-parallel systems using a column generation approach.* IEEE Transactions on Reliability, 2010. **59**(4): p. 706-717.
9. Tanner, D.M., et al. *The effect of frequency on the lifetime of a surface micromachined microengine driving a load*. in *Reliability Physics Symposium Proceedings, 1998. 36th Annual. 1998 IEEE International*. 1998.
10. Tanner, D.M., N.F. Smith, and D.J. Bowman, *First reliability test of a surface micromachined microengine using SHiMMeR.* Proc. SPIE, 1997. **3224,**.
11. Hartzell, A.L., et al., *Lifetime Prediction MEMS Reliability*. 2011, Springer US. p. 9-42.

