Developing a New Model for availability Optimization Applied to a Series-Parallel System

A. Yahyatabar Arabi, A. Eshraghiaye Jahromi* & M. Shabannataj

Available online at www.SID.ir

ABSTRACT

Redundancy technique is known as a way to enhance the reliability and availability of non-reparable systems, but for repairable systems, another factor is getting prominent called as the number of maintenance resources. In this study, availability optimization of series-parallel systems is modelled by using Markovian process by which the number of maintenance resources is located into the objective model under constraints such as cost, weight, and volume. Due to complexity of the model as nonlinear programming, solving the model by commercial softwares is not possible, and a simple heuristic method called as simulated annealing is applied. Our main contribution in this study is related to the development of a new availability model considering a new decision variable called as the number of maintenance resources. A numerical simulation is solved and the results are shown to demonstrate the efficienc of the method.

KEYWORDS

Availability optimization, Maintenance resource, Redundancy level

1. Introduction

Evaluate the performance of repairable system such as manufacturing systems, power industry, aircraft and oil industry. Two general ways has been developed to increase availability of an engineering system known as increasing the availability of each component and redundancy level. Availability of a system directly depends on availability of its components. Reliability and maintainability of each components lead to enhancement in availability. Due to limitation in technology, the second way is redundancy allocation. Redundancy in a system means that the components are structured in parallel. Redundancy allocation problem (RAP) is the most common method to meet the optimization of reliability and availability subject to the realistic constraints such as cost, weight, volume, etc [4, and 8]. Considering nature and application of systems, they are categorized into repairable and non-repairable systems. In the real world, so many systems are repairable and a repairable system can be restored and work properly. Repairable series-parallel systems are frequently used in practice, e.g., power systems, telecommunications systems, manufacturing production systems, and industrial systems. In this regard, availability is getting more serious. A comprehensive study on repairable system is presented in [11, and 12]. Barlow and Hunter [1] first presented a minimal repair model in which the minimal repair does not affect the lifetime of the system. A cold stand-by repairable system has been studied by [14]. Each component after repair is not ‘as good as new’. With respect to this assumption, a geometric process was used. Baron et al. [2] investigates a k-out-of-n system with several predetermined repair facilities in which phase type repair time and both strategies are considered. Vanderperre [13] studies on a system consisting of several active and standby components with constant failure rate and arbitrary distribution for repair time. A single repair facility is considered. In
A. Yahyatabar Arabi, A. Eshraghiaye Jahromi & M. Shabannatj  Developing a New Model for availability … 102

[6], Frostige et al. propose a study on the availability of a k-out-of-n repairable system. They use Markov renewal processes for a k-out-of-n system in which components are repairable. Zhang et al. [15] analyze a k-out-of-(m+n) warm standby repairable system. The system is divided into two separate types of components, m and n components. The first type is supposed to have lower failure rate. In accordance with the Markov process, a method is proposed to find the solutions of system availability. Levitin and Lisiński [9] presented an optimization method that minimizes the total cost considering replacement frequency, preventive and corrective maintenance. Elegbede and Adjallah [5] proposed a method based on Genetic Algorithm and experiments plan to optimize the availability and the cost of repairable parallel-series systems.

Castro and Cavalcá [3] studied on availability optimization of series-parallel systems considering redundancy level and team maintenance action as decision variables considering constraints such as cost, weight, and volume. Castro and Cavalcá [3] applied dependability ratio to consider maintenance team as a decision variable in objective function. Genetic algorithm was the method to solve their problem. The impact index “I” has been introduced as an input parameter to the model, defined as the dependability ratio sensitivity to the corrective maintenance resources on a specific component.

The aim of this paper is to present a new model for availability optimization of series-parallel system using Markovian process considering common constraints such as cost, weight, and volume. Due to complexity degree of the model, a simulated annealing method is proposed to solve the model. Simulated annealing as a straightforward heuristic algorithm is used to solve the proposed problem of the new model. In this study, our main contribution is that maintenance resources are taken account into the objective function without any additional input, vital parameters such as parameter in [3] that its value significantly affects the solution. In [3], an input parameter is added to the model called as impact index “I” affecting obtained solutions, to properly determine the value of “I” is vital to get the optimal or near optimal solution. This paper organized as follows. Section 2 presents a proposed model and mathematical formulation. The simulated annealing method and its application is described in section 3. An adapted example is solved in section 4. Finally, section 5 presents conclusion.

2. The Proposed Availability Optimization Problem

2-1. System Description

This study proposes an availability model for a series system with multiple redundant subsystems including repairable active components. In this model, to increase the availability of system, two decision variables are considered: the number of maintenance resources and redundancy level. The goal of the problem is to find the optimal values of these decision variables in each subsystem in the presence of the constraints such as weight, cost, and volume. The notations and assumption of the model are presented in the following.

2-1-1. Notation

\( m \): the number of subsystem

\( n_j \): the number of redundant components in the subsystem \( j \)

\( \lambda_i \): failure rate of each component when there are \( i \) failed component

\( \lambda_j \): failure rate of each component in the subsystem \( j \)

\( \mu_i \): repair rate, when there are \( i \) failed component

\( \mu_j \): repair rate of each component in the subsystem \( j \)

\( r_j \): the number of maintenance resource allocated to the subsystem \( j \)

\( P_{i,j}(t) \): the probability that there are \( i \) failed components in the subsystems \( j \) at time \( t \)

\( P_{i,j}(t) \): steady-state probability that there are \( i \) failed components in the subsystem \( j \)

\( P_{i,j}(t) \): first derivation of \( P_{i,j}(t) \)

\( A_{sub,j}(t) \): point availability of the subsystem \( j \) at time \( t \)

\( A_{sub,j} \): steady-state availability of the subsystem \( j \)

\( A_j \): availability of the system

\( C \): system cost

\( W \): system weight

\( V \): system volume

\( CM \): system maintenance cost

\( c_j \): component cost in the subsystem \( j \)

\( w_j \): components weight in the subsystem \( j \)

\( v_j \): component volume in the subsystem \( j \)

\( q_j \): probability of failure in the subsystem \( j \)

2-2-1. Assumptions

- Only one maintenance resource is allowed to allocate to the repair of a failed component. The time to repair a failed component follows identical independent exponential. \( (1 \leq r_j \leq n_j) \).
- All components in each subsystem follow independent exponential lifetime distribution.
- Subsystem \( j \) is called failed as soon as the number of failed component just reached \( n_j \), and the
system is considered failed as soon as one of the subsystem has failed.

- When a component of a subsystem fails, allocated repairman of the subsystem, if available, immediately begins; if not, the failed component must wait for the repairman. The repair is based on first-come, first-served.

- The probability that two or more components are restored or become failed simultaneously in a small interval is not considered.

2.2. Mathematical Model

This model presents an objective function associated to the maximization of the availability. The equation (1) can express the availability of the system regarding Fig. 1 as follows:

\[ A_j = \prod_{i=1}^{m} A_{subj} \]  

(1)

\[ P_{i,j}(t+\Delta t) \] represents the probability that there are \( i \) failed components in the subsystem \( j \) at time \( (t, t+\Delta t) \). To obtain this probability, following events are evaluated:

During \( (t, t+\Delta t) \), a transition happens, the subsystem \( j \) is in status \( i+1 \) at time \( t \) and is in status \( i \) at time \( t+\Delta t \).

During \( (t, t+\Delta t) \), a transition happens, the subsystem \( j \) is in status \( i-1 \) at time \( t \) and is in status \( i \) at time \( t+\Delta t \).

During \( (t, t+\Delta t) \), no change in the subsystem \( j \) status happens.

During \( (t, t+\Delta t) \), the subsystem \( j \) status is transmitted by two or more.

According to the Poisson process, the last event is negligible and close to zero. Based on the Fig. 1, the probability of the subsystem can be indicated in the following:

\[ P_{i,j}(t+\Delta t) = P_{i+1,j}(t)(1 - \lambda_{i+1}\Delta t)(\mu_{i+1}\Delta t) + \\
\lambda_{i+1} P_{i+1,j}(t)(1 - \lambda_{i+1}\Delta t)(1 - \mu_{i+1}\Delta t) + \\
P_{i,j}(t)(1 - \lambda_{i}\Delta t)(1 - \mu_{i}\Delta t) + \\
P_{i-1,j}(t)(\lambda_{i} P_{i-1,j}(t)(1 - \mu_{i}\Delta t) + \\
P_{i,j}(t)(1 - \lambda_{i+1}\Delta t)(1 - \mu_{i}\Delta t) + \\
P_{i+1,j}(t)(\lambda_{i+1}\Delta t)(\mu_{i+1}\Delta t) + \\
P_{i,j}(t)(\lambda_{i+1}\Delta t)(\mu_{i+1}\Delta t) \]  

(4)

The primary stage of making the model is to calculate the availability of the subsystem \( j \) that is explained in the following:

\[ A_{subj}(t) = 1 - P_{n-j-1,1,j}(t) \]  

(2)

\[ A_{subj} = \lim_{t \to \infty} A_{subj}(t) = 1 - \lim_{t \to \infty} P_{n-j-1,1,j}(t) \]  

(3)

As per the model description, steady-state availability of a parallel system is described using Markovian process to obtain the availability of each subsystem. The state transition diagram of a parallel system is given in Fig. 1. The number in each circle states the number of failed components.

Fig. 1. The state transition diagram of a repairable parallel system

Where, \( \Delta t \to 0 \)

\[ P_{i,j}(t) = -(\lambda_{i} + \mu_{i}) P_{i,j}(t) + \lambda_{i} \mu_{i+1} P_{i+1,j}(t) + \]  

(5)

\[ P_{i,j} = \lim_{t \to \infty} P_{i,j}(t) \]  

(6)

\[ \lim P_{i,j}(t) = \lim(-\lambda_{i} + \mu_{i}) P_{i,j}(t) + \\
\lambda_{i+1} P_{i+1,j}(t)(1 + \mu_{i+1} P_{i+1,j}(t)) \]  

(7)

\[ -(\lambda_{i} + \mu_{i}) P_{i,j} + \lambda_{i+1} P_{i+1,j} + \mu_{i+1} P_{i+1,j} = 0 \]  

(8)

\[ -\lambda_{0} P_{0,j} + \mu_{1} P_{1,j} = 0 \]  

(9)

\[ P_{1,j} = \frac{\lambda_{0}}{\mu_{1}} P_{0,j} P_{2,j} = \frac{\lambda_{0}\lambda_{1}}{\mu_{1}\mu_{2}} P_{0,j} \]  

(10)

\[ P_{i,j} = \frac{\lambda_{0}...\lambda_{i-1}}{\mu_{1}...\mu_{i}} P_{0,j} \]  

\[ P_{0,j} + P_{1,j} + ... + P_{n-j,j} = 1 \]  

(11)

\[ P_{0,j} = (1 + \sum_{i=1}^{n-j} \frac{\lambda_{0}...\lambda_{i-1}}{\mu_{1}...\mu_{i}})^{-1} \]  

(12)
To evaluate the availability of subsystem based on equations above:

\[ A_{sub \ j} = 1 - P_{nj - k_j + k_j} = 1 - \frac{\lambda_{nj - k_j}^{j} \cdots \lambda_{nj}^{j}}{\mu_{n-1} \cdots \mu_{1}} (1 + \sum_{i=1}^{n_j - k_j + 1} \frac{\lambda_{nj - k_j}^{j} \cdots \lambda_{n_i}^{j}}{\mu_{i-1} \cdots \mu_{1}}) \]

Therefore the availability of the system based on equation (1) can be represented as follows:

\[ A_j = \prod_{j=1}^{m} (1 - \frac{n_j!}{r_j! r_j^{n_j}} \left( \frac{\lambda_j}{\mu_j} \right)^{n_j}) 
\]

\[ ((1 + \sum_{i=1}^{r_j} \frac{n_j!}{(n_j - i)! r_j! r_j^{n_j - i}} \left( \frac{\lambda_j}{\mu_j} \right)^{n_j - i}))) \]

This study uses the result of queuing theory for the failure rate of the subsystem \( j \) and repair rate of the subsystem \( j \) as follow:

\[ \lambda_j = (n-j)\lambda \]  

\[ \mu_i = \begin{cases} i\mu & 0 \leq i \leq r_j \\ r_i \mu & r_j \leq i \leq n_j \end{cases} \]

According to equations above, the mathematical model of the system can be computed as below:

Max

\[ A_j = \prod_{j=1}^{m} (1 - \frac{n_j!}{r_j! r_j^{n_j}} \left( \frac{\lambda_j}{\mu_j} \right)^{n_j}) 
\]

\[ ((1 + \sum_{i=1}^{r_j} \frac{n_j!}{(n_j - i)! r_j! r_j^{n_j - i}} \left( \frac{\lambda_j}{\mu_j} \right)^{n_j - i}))) \]

\[ \lambda_j = (n-j)\lambda \]  

\[ \mu_i = \begin{cases} i\mu & 0 \leq i \leq r_j \\ r_i \mu & r_j \leq i \leq n_j \end{cases} \]

Fitness function is the main part of all heuristic algorithms. Fitness function plays a key role to guarantee the feasible and global solution. With respect to equations (18) to (22), the fitness function in the proposed algorithm is represented as follows:

\[ F = A_j / (1 - \min(0, (C - \sum_{j=1}^{m} (c_j n_j + h_j r_j))) + \min(0, (CM - \sum_{j=1}^{m} (w_j n_j + w_j n_j))) + \min(0, (W - \sum_{j=1}^{m} (w_j n_j))))) \]
4. A Numerical Example

To illustrate the analysis of the proposed problem, a numerical example is solved by the simulated annealing method. The input data is presented in Table 1. Time T is set to 100 time units. A system with five subsystem has been solved by the simulated annealing described in the previous section. By using MATLAB R2009ea on the Intel Core 2 Duo CPU 2.66 and GHz 2.67 GHz PC, an algorithm has been coded to find the solution. Due to the stochastic nature of the proposed SA, 50 independent runs were made for the example involving 50 different initial solutions. For all examples, the parameters of SA are set as follows: \( \alpha \) (cooling rate)=0.9, \( T_0=1000 \), \( T_f \) (the final temperature)=0.01, Miter=200. The maximum availability found by the proposed SA is used to compare the exact solution. The result of running the SA is presented in Table 2. It is obviously shown that, the difference between the solution by the proposed algorithm and the exact optimal solution is so trivial.

**Tab. 1. System data**

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>( \lambda )</th>
<th>( \mu )</th>
<th>Design Cost</th>
<th>Weight</th>
<th>Volume</th>
<th>Corrective Maintenance Cost</th>
<th>Corrective Maintenance Resource Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.002</td>
<td>0.02</td>
<td>50</td>
<td>50</td>
<td>55</td>
<td>60</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>0.0018</td>
<td>0.028</td>
<td>55</td>
<td>45</td>
<td>50</td>
<td>40</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>0.0016</td>
<td>0.025</td>
<td>55</td>
<td>80</td>
<td>70</td>
<td>45</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>0.0013</td>
<td>0.033</td>
<td>40</td>
<td>35</td>
<td>35</td>
<td>30</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>0.002</td>
<td>0.033</td>
<td>60</td>
<td>70</td>
<td>65</td>
<td>50</td>
<td>5</td>
</tr>
</tbody>
</table>

**Tab. 2. Comparison between the proposed method and exact optimal solution**

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Redundancy level</th>
<th>Maintenance resource</th>
<th>Redundancy level</th>
<th>Maintenance resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

**Tab. 3. Statistical results of running SA for 50 runs**

<table>
<thead>
<tr>
<th></th>
<th>Minimum</th>
<th>Maximum</th>
<th>Mean</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.980789039</td>
<td>0.990985041</td>
<td>0.99065441</td>
<td>0.001381065</td>
</tr>
</tbody>
</table>

Fig. 2 illustrates the convergence of the fitness for the example, the near optimal solution is approximately achieved after 20 generations. Maximum, minimum, average and standard deviation of all examples are shown in table 3. It is obviously indicated that all standard deviations are trivial and difference between maximum and minimum is also trivial in each example. The results in table 3 indicate that the proposed algorithm strongly converged to the optimal solution.

5. Conclusion

In this paper, availability optimization of a series system with five redundant subsystems has been modelled through Markovian process. The main contribution of this study refers to consider the number of maintenance resources and redundancy level together as decision variables subject to constraints such as weight, cost, and volume.

The presented model has been formulated as nonlinear integer programming. To find the exact solution of the problem is not easy especially for large systems. A heuristic method called as simulated annealing used to solve the problem. Results calculated
in table 2 and 3 shows the proper performance of the model and method.

Reference


