Reliability evaluation of non-reparable three-state systems using Markov model and its comparison with the UGF and the recursive methods

Pedram Pourkarim Guilani[a], Mani Sharifi[a], S.T.A. Niaki[b1], Arash Zaretalab[a]

[a] Faculty of Industrial & Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

[b] Department of Industrial Engineering, Sharif University of Technology, P.O. Box 11155-9414 Azadi Ave., Tehran 1458889694, Iran

Abstract

In multi-state systems (MSS) reliability problems, it is assumed that the components of each subsystem have different performance rates with certain probabilities. This leads into extensive computational efforts involved in using the commonly employed universal generation function (UGF) and the recursive algorithm to obtain reliability of systems consisting of a large number of components. This research deals with evaluating non-repairable three-state systems reliability and proposes a novel method based on a Markov process for which an appropriate state definition is provided. It is shown that solving the derived differential equations significantly reduces the computational time compared to the UGF and the recursive algorithm.

Keywords: Reliability; Multi-state systems; Three-state components; UGF; Markov process, Recursive algorithm

[1] Corresponding author, Email: niaki@sharif.edu

1. Introduction

In the classical system reliability viewpoint, systems are binary-state, in which the components have two states of "working perfect" or "completely failed." Whereas according to newer approaches, a typical component of a system may work at any performance rate between 0% and 100%, each with a certain probability. These systems that are called multi-state (MSS) have been studied in depth by many researchers such as Lisnianski & Levitin (2003) and Zuo & Tian (2009). While the binary-state system reliability can be obtained through the basic mathematical and statistical relations, most of the research works in MSS reliability problem focus on optimizing the level of a unique system redundancy (Lisnianski & Levitin, 2003).

System reliability determination of MSSs is extremely hard using mathematical relations (if not possible), because the system states increase rapidly and computational complexity gets high. As an alternative, Ushakov (1986) proposed the universal generation function (UGF) approach for the first time. UGF is known as an appropriate method for calculating the reliability and availability of multi-state systems. This method incredibly decreases the number of system state evaluations and makes the system reliability and availability computations easier (Ding & Lisnianski 2008). Later, Gnedenko & Ushakov (1995), Ushakov (2000), and Lisnianski & Levitin (2003) introduced more applications of the UGF method. Lisnianski et al. (1996) utilized UGF to evaluate the reliability of a MSS containing serial, parallel, and series-parallel sub-systems. Ding & Lisnianski (2008) showed that the output probability distribution for the entire MSS could be determined by UGF. Moreover, Levitin et al. (1998), Levitin & Lisniaski (1998), and Lisnianski et al. (2000) calculated the distribution function of MSSs with series, parallel, and bridge structures by combination of different operators. Further, Levitin et al. (1998) presented a redundancy optimization algorithm for a MSS with series-parallel structure using UGF. Kuo &

Wan (2007) discussed the optimal reliability design in which UGF was employed as the main method in appraising multi-state systems reliability evaluation. Besides, UGF was applied in redundancy allocation problems (RAP) and k-out-of-n systems many times (see for example Tian et al., 2009; Levitin & Lisnianski, 2003; Ouzineb et al., 2008).

Although UGF is known as a convenient method in order to calculate MSS reliability, because it can evaluate reliability of multi-state systems with series, parallel, series- parallel, and bridge structures ((Levitin et al. (1998); Lisnianski et al. (2000); Levitin & Lisnianski (1998)), but when the number of components in the system increases the required CPU time dramatically increases. This is the main pitfall for which some solutions have been proposed in the literature. Wu & Chen (1994) offered a recursive algorithm in order to evaluate the reliability of a binary weighted k-out-of-n system. Higashiyama (2001) proposed a procedure in order to evaluate the system reliability of a binary weighted k-out-of-n system in fewer time in comparison with the proposed method by Wu & Chen (1994). However, the time and the space complexities of these two methods are similar to the ones of a recursive algorithm proposed by Li & Zuo (2008), which is another useful method to evaluate system availability of binary weighted k-out-of-n:G as well as multi-state weighted k-out-of-n:G systems with less required CPU time. Nevertheless, this method cannot be used when the number of components gets large.

In this paper, a novel and effective approach is aimed to calculate system reliability of non-repairable three-state systems by first defining an appropriate system state. Then, using a Markov process and utilizing the Chapman-Kolmogorov theorem (Newell 1982), differential equations are obtained. We show that solving these differential equations, which leads into state probabilities and consequently the system reliability, requires much less CPU time and provides solutions with the same quality compared to the ones obtained by either UGF or the recursive algorithm. To do this, a brief background on the UGF method is first presented in Section 2. Then, the recursive algorithm is described briefly in Section 3. Next, the problem is defined, the proposed Markov process to model the problem is introduced, and the differential equations are derived in Section 4. Section 5 concerns with evaluating and comparing the performances of the proposed method. Conclusion and recommendations for future research come in Section 6.

2. The universal generation function (UGF)

In order to describe the UGF method, consider a system having n components, where components $j\ ;\ j=1,2,...,n$, may have k_j different states with certain probabilities of performance rates denoted by an ordering set $g_j=\left\{g_{j1},g_{j2},...,g_{jk_j}\right\}$ in which g_{ji_j} represents the performance rate of component j in state $i_j \in \{1,2,...,k_j\}$. The performance rate $G_j(t)$ of component j at time $t \geq 0$ is a random variable taking values in $g_j : G_j(t) \in g_j$. Moreover, let the probabilities associated with different states of component j be the set $P_j=\left\{p_{j1},p_{j2},...,p_{jk_j}\right\}$ (Lisnianski & Levitin, 2003). Furthermore, $g_{ji_j} \rightarrow P_j$ is often called the probability mass function (pmf) (Levitin, 2005; Ding & Lisnianski, 2008).

As soon as the performance rates of the components are given, the performance rate of a MSS can be determined. Let the system have K different states and g_i be the performance rate of the system in state $i\ ;\ i=1,2,...,K$. Then, the system performance rate at time $t \geq 0$ will be either a random variable or a random vector that takes values in $\left\{g_1,...g_i,...,g_K\right\}$. Thus, the space representing all possible combinations of performance rates for all components is $L^n=\left\{g_{11},...,g_{1k_1}\right\}\times...\times\left\{g_{j1},...,g_{jk_j}\right\}\times...\times\left\{g_{n1},...,g_{nk_n}\right\}$ and the space for all possible values of the entire system performance rates is $M = \{g_1, ..., g_K\}$. The transformation $\phi\left(G_1(t)...G_n(t)\right): L^n \to M$ that maps the space of performance rates of system components into the space of system performance rates is called the system structure function (Lisnianski and Levitin, 2003). Moreover, the total number of possible states (performance rates) of a MSS is

$$K = \prod_{j=1}^{n} k_j \tag{1}$$

Besides, the probability associated with the state i of the system can be obtained as

$$P_i = \prod_{j=1}^{n} p_{ji_j} \tag{2}$$

Denoting the MSS performance rate for state i as

$$g_i = \phi\left(g_{1i_1}, g_{2i_2}, ..., g_{ni_n}\right) \tag{3}$$

the probability distribution of the whole system for K combinations of $i_1, i_2, ..., i_n$ is

$$g_i = \phi\left(g_{1i_1}, g_{2i_2}, ..., g_{ni_n}\right) \ ; \ P_i = \prod_{j=1}^{n} p_{ji_j} \tag{4}$$

where $1 \le i_j \le k_j, \left(1 \le j \le n\right)$.

The z-transform of a random variable $G_j(t)$ represents its pmf with $p_j = \left\{p_{j1}, p_{j2}, ..., p_{jk_j}\right\}$ associated with $g_j = \left\{g_{j1}, g_{j2}, ..., g_{jk_j}\right\}$ (Lisnianski & Levitin, 2003; Levitin 2005). Equation (5) shows the probability distribution of the component j, called also individual UGF.

$$u(z) = \sum_{i=1}^{k_j} p_{ji_j} \cdot z^{g_{ji_j}} \tag{5}$$

To derive the probability distribution of the entire MSS with an arbitrary structure function ϕ, a general composition operator Ω_ϕ is employed on individual UGF of n components as (Lisnianski & Levitin, 2003):

$$U(z) = \Omega_\phi\{u_1(z), \ldots, u_n(z)\} = \Omega_\phi\left\{\sum_{i_1=1}^{k_1} p_{1i_1} \cdot z^{g_{1i_1}}, \ldots, \sum_{i_n=1}^{k_n} p_{ni_n} \cdot z^{g_{ni_n}}\right\}$$
$$= \sum_{i_1}^{k_1}\sum_{i_2}^{k_2}\cdots\sum_{i_n}^{k_n}\left(\prod_{j=1}^{n} p_{ji_j} \cdot z^{\phi\left(g_{1i_1},\ldots,g_{ni_n}\right)}\right) \quad (6)$$

Based on the relationship between MSS performance and the demand level ω that is often determined outside the system, the state space of a MSS can be divided into two subsets: acceptable and unacceptable. The relationship usually is determined by the system state adequacy index r_i defined by $r_i = g_i - \omega$. As a result, state i is acceptable if $r_i \geq 0$. The availability of a MSS (reliability of a non-repairable MSS) is defined as the probability the system staying in the subset of acceptable states. Thus, based on the demand level ω the availability of a MSS, $A(\omega)$, is usually defined as the probability the MSS performance rate is greater than ω (Lisnianski and Levitin, 2003). In other words,

$$A(\omega) = \sum_{r_i \geq 0} p_i \quad (7)$$

Then, using operator δ_A it becomes

$$A(\omega) = \delta_A\left(U(z), \omega\right) = \delta_A\left(\sum_{i=1}^{K} p_i \cdot z^{g_i}, \omega\right) = \sum_{i=1}^{K} p_i \cdot \alpha_i$$

where (8)

$$\alpha_i = \begin{cases} 1 \ , \ r_i \geq 0 \\ 0 \ , \ r_i < 0. \end{cases}$$

In Eq. (8) δ_A is known as UGF operator. This operator determines the polynomial UGF for a group of components first connected in parallel in a subsystem and then for a group of subsystems in series using simple algebraic operations on the individual UGF of components. In some cases, composition operators can be developed for structures with more complex system structure, such as bridges, as shown by Levitin and Lisnianski (2000)).

3. The recursive algorithm

Li and Zuo (2008) presented a recursive algorithm in order to evaluate reliability of multi-state weighted k-out-of-n:G systems. In this type of systems, each component is classified to work in different states and the system is working until sum of the weight of the safe components is at least k. Li & Zuo (2008) showed that their method is capable to calculate system reliability in less CPU time compared to UGF.

The notations used in Li and Zuo (2008) in their method are:

n : The number of components in each system,

M : The highest possible state of each component and system,

w_{ij} : The weight of component i when it is in state j,

p_{ij} : The probability of component i to be in state j,

q_{ij} : The probability of component i to be in states bellow j, $q_{ij} = \sum_{i=0}^{j-1} p_{i,l}$,

k_j : The minimum total weight required to ensure that the system is in states greater than or equal j,

$R_j(k_j, n)$: Probability that the system is in state j or greater, The recursive equation for evaluation of the state distribution of the system is as follows:

$$R_j(k_j, i) = \sum_{r=0}^{r=M} p_{i,r}.R_j(k_j - w_{i,r}, i-1) \tag{9}$$

where the boundary conditions are given in Eq. (9).

$$\begin{aligned} &R_j(k, 0) = 0 \quad \text{when} \quad 0 < k \le k_j \\ &R_j(k, i) = 1 \quad \text{when} \quad i \ge 0 \ \text{ and } k \le 0 \end{aligned} \tag{10}$$

Interested readers are referred to Li and Zuo (2008) for more details on the recursive algorithm.

4. Problem and state definition

Consider a system consisting of s sub-systems in series shown in Fig. 1, where each sub-system involves n_i parallel components. Moreover, each component has one of the following performance states at any time:

1) Working with full performance (%100 performance),
2) Working with half or semi performance (%50 performance),
3) Not working or failed.

A sub-system is safe unless all of its components fail. It is assumed that components are non-repairable and also the components work under a constant failure rate (CFR) condition. Moreover, the components of the *i*th sub-system have three different failure rates as

λ_{i1}: Transforming from 100% working (full performance) to 50% working (semi performance)

λ_{i2}: Transforming from 100% working to 0% working performance

λ_{i3}: Transforming from 50% working to 0% working performance

Insert Figure 1 about here

To distinct between sub-systems with components that operate in different working states, fully working components receive two points (weights) each, semi working components receive one point and failed components receive no point. For instance, if in a three-component sub-system, two components are working with full performance and one component is working with semi performance, the sub-system earns 5 points. Hence, the possible points of the *i*th sub-system with n_i components are

$$\left(2n_i\,, 2n_i-1, 2n_i-2, \ldots, 2, 1, 0\right) \tag{11}$$

The state transition of a single three-state component is presented in Fig. 2. Furthermore, based on the above pointing scheme since a three-component sub-system with one full-performance component and two failed ones and another one with zero full-performance and two semi-performance components both receives 2 points, the notation $\left\{(w, m) \quad ; \quad w, m < n\right\}$ is presented for a sub-system with n component in which w and m are the numbers of full-performance and semi-performance components, respectively. As a result, the i^{th} sub-system can receive k points where $k = 0, 1, 2, \ldots, 2n_i - 1, 2n_i$. Furthermore, the number of system states can be obtained by:

$$\begin{array}{l} 2w + m = k \quad ; \quad k = 0, 1, 2, \ldots, 2n_i - 1, 2n_i \\ w + m \le n \end{array} \tag{12}$$

Insert Figure 2 about here

4.1. State space diagram and differential equations

Consider a three-state system with s sub-systems and states $\{(w,m)\ ;\ w,m<n\}$, described in Section 3. The state space of each sub-system is shown in Figure 3. According to Markov property we know that $p'_{(w,m)}$ = [inflow to state (w,m)] - [outflow to state (w,m)]. Then, considering this fact that the components of each sub-system may have three different performances at any time, each state can be obtained from one of the previous states or can be transformed to the next states with related failure rate if it is possible (see Fig. 3). In other words, considering the failure rates of each component as parameters of exponential distributions, the states $(w+1,m-1)$, $(w+1,m)$ and $(w,m+1)$ can transform themselves into (w,m) with failure rates of $(w+1)\lambda_1$, $(w+1)\lambda_2$, and $(m+1)\lambda_3$, respectively. Also the state (w,m) can be transformed into $(w-1,m+1)$, $(w-1,m)$, and $(w,m-1)$ with failure rates $w\lambda_1$, $w\lambda_2$, and $m\lambda_3$, respectively.

Insert Figure 3 about here

The differential equation of each sub-system can be obtained based on Fig.3 using the inflow and outflow of any state. Since the components can be in three different conditions (number of fully, semi, and failed components) at any time, the numbers of possible inflow and outflow are different. Therefore, the differential equation of any state may include full or part of follow relation shown in Eq. (13).

$$\begin{cases} P'_{(n,0)}+\left(n\lambda_1+n\lambda_2\right)P_{(n,0)}=0 & :\ w=n,\ m=0 \\ P'_{(w,m)}+\left(w\lambda_1+w\lambda_2+m\lambda_3\right)P_{(w,m)}= & \\ \left(w+1\right)\lambda_1P_{(w+1,m-1)}+\left(w+1\right)\lambda_2P_{(w+1,m)}+\left(m+1\right)\lambda_3P_{(w,m+1)} & :\ w,m<n \end{cases} \tag{13}$$

To make the equations easier to solve, a matrix model is first proposed. Then, the solution of the system of differential equations leads into the probability of each state. As a result, the reliability of sub-system i becomes:

$$R_i(t) = \sum_{(w,s)\in R_{[W,S]}-(0,0)} P_{(w,s)}(t) \tag{14}$$

The steps involved in evaluating non-repairable three-state systems reliability using the proposed method are shown in Fig. 4.

Insert Figure 4 about here

As an example, consider a sub-system with three identical components. Then, the number of states can be obtained by:

$$\begin{aligned} &2w + m = k \quad ; \quad k = 0, 1, 2, \ldots, 6 \\ &w + m \le 3 \end{aligned} \tag{15}$$

Equation (15) can be decomposed into seven equations as follows:

$$\begin{aligned}
&k = 6 \Rightarrow \begin{Bmatrix} 2w + m = 6 \\ w + m \le 3 \end{Bmatrix} \Rightarrow (w, m) = \{(3,0)\} \\
&k = 5 \Rightarrow \begin{Bmatrix} 2w + m = 5 \\ w + m \le 3 \end{Bmatrix} \Rightarrow (w, m) = \{(2,1)\} \\
&k = 4 \Rightarrow \begin{Bmatrix} 2w + m = 4 \\ w + m \le 3 \end{Bmatrix} \Rightarrow (w, m) = \{(2,0), (1,2)\} \\
&k = 3 \Rightarrow \begin{Bmatrix} 2w + m = 3 \\ w + m \le 3 \end{Bmatrix} \Rightarrow (w, m) = \{(0,3), (1,1)\} \\
&k = 2 \Rightarrow \begin{Bmatrix} 2w + m = 2 \\ w + m \le 3 \end{Bmatrix} \Rightarrow (w, m) = \{(1,0), (0,2)\}
\end{aligned} \tag{16}$$

$$k = 1 \Rightarrow \begin{Bmatrix} 2w + m = 1 \\ w + m \le 3 \end{Bmatrix} \Rightarrow (w, m) = \{(0,1)\}$$

$$k = 0 \Rightarrow \begin{Bmatrix} 2w + m = 0 \\ w + m \le 3 \end{Bmatrix} \Rightarrow (w, m) = \{(0,0)\}$$

Hence,

$$U_{(w,m)} = \{(0,0),(0,1),(0,2),(0,3),(1,0),(1,1),(1,2),(2,0),(2,1),(3,0)\} \quad (17)$$

The space state diagram and the matrix model of this example are presented in Fig. 5 and Table 1, respectively.

Insert Figure 5 about here

Insert Table 1 about here

5. A comparison study

In order to demonstrate the validity of the proposed method and to compare its solution as well as its speed to find the solution to the ones of the well-known UGF method and the recursive algorithm, consider a non-repairable system with six sub-systems that are connected serially. The triple failure rates of the components are shown in Table 2. A non-repairable system is assumed because its reliability and availability both refer to the same concept (Tian et al.; 2009).

Insert Table 2 about here

To calculate the system reliability using the UGF method, each component of the system is assumed three performance states as 0, 1, and 2 for failed, semi-working, and fully working, respectively. The probabilities associated with these three states are obtained through the following system of differential equations:

$$\begin{aligned} P_2'(t) &= -(\lambda_1 + \lambda_2)P_2(t) \\ P_1'(t) &= \lambda_1 P_2(t) - \lambda_3 P_1(t) \\ P_0'(t) &= \lambda_2 P_2(t) + \lambda_3 P_1(t) \end{aligned} \tag{18}$$

Moreover, in order to evaluate the reliability of each sub-system using the recursive algorithm, the following notations are used:

n_s : The number of components in each sub-system,

M : The highest possible state (or point) of each component (it is equal to two in this paper for components with a fully working performance),

w_{ij} : The weight (or point) of component *i* when it is in state *j*. In this paper, it is 0, 1, and 2 for failed, semi-working, and fully working,

p_{ijs} : The probability of component *i* to be in state *j* in sub-system *s*. These values are obtained by solving the system of differential equations in (18),

k_j : The minimum total weight (or point) required to ensure that the system is in states greater than or equal *j*. (As in this paper it is assumed that each sub-system is safe until there is at least one component as semi working, j is equal to 1 and $k_1 = 1$),

$R_j(k_j, n)$: Probability that the sub-system with n_s components is in state *j* or greater,

Based on the above notations, the recursive equation to evaluate state distribution of each sub-system is as follows:

$$R_{sj}(k_1, i) = \sum_{r=0}^{r=M} p_{i,r,s}.R_{sj}(k_1 - w_{i,r}, i-1) \tag{19}$$

In which:

$M = 2$
$k_1 = 1$
$w_{i1} = 0 \;\; , \; w_{i2} = 1 \;\; , \;\; w_{i3} = 2 \qquad \forall i$

Moreover, the boundary conditions for the presented recursive algorithm are:

$$R_j(k,0) = 0 \quad \text{when} \quad 0 < k \le k_j$$
$$R_j(k,i) = 1 \quad \text{when} \quad i \ge 0 \;\; \text{and}\; k \le 0 \tag{20}$$

Finally, the system reliability is obtained by multiplying the reliability of each sub-system.

The comparisons are made based on different numbers of components in each of the six sub-systems using a mission time of 100 hours. The computer programs for all the three approaches are developed in MATLAB, version 7.10.0.499. To run these programs, a Pentium 4 computer with a core 2 CPU 2.4 GHZ and 3GB RAM under Windows 7 operating system are used. The results of the comparisons are shown in Table 3. Moreover, the system structure of the 6th presented example in Table 3 ([3 3 3 4 4 4]) is shown in Fig .6.

Insert Table 3 about here

Insert Figure 6 about here

The results in Table 3 show that the obtained three-state system reliabilities of the three methods are almost the same for all sub-systems. However, as the sub-systems becomes large in terms of its number of components, while the computational time of the UGF method increases dramatically, the computational time of the recursive algorithm is significantly bigger than the one of the proposed method. Nevertheless, the results show that the computational time of the proposed method is not affected by the size of the sub-system very much. This shows the higher efficiency of the proposed method in terms of the computational time compared to the previous methods. Note also that the UGF method due to its inherent exhaustive enumeration property is not able to obtain reliabilities of five and the six-component sub-systems in a reasonable time. This is due to the fact the UGF method with information about performance rates and theirs relative probabilities checks all of possible states, classifies the states based on system structure, and evaluate the reliability or availability of the system. That is why it is difficult (or even impossible) to calculate system reliability in reasonable time when the number of components and states gets large. To show this better, Fig.7 illustrates a graphical comparisons of the required CPU times shown in Table 3. In this figure, horizontal pivot represents experiment types and vertical pivot represents time duration to complete the experiment. As Fig. 7 shows, the computational time of the UGF method is significantly larger than the ones of the other two methods. Note in this figure that as the difference between the speed of the proposed method and the recursive algorithm is not clear in a small scale, the scale has been enlarged to show the difference clearly.

Insert Figure 7 about here

In order to demonstrate the good performance of the proposed method in large-scale systems, some examples are solved here. In these examples, it is assumed that the failure rates are randomly generated from a uniform distribution over (0.001, 0.01), where the performance of proposed method is compared with the one of the recursive algorithm. Furthermore, the number of components is assumed to be kept equal in all sub-systems. The results are demonstrated in Table 4. According to Table 4, it is clear that the proposed method has a significant advantage in terms of the computational time over the recursive algorithm in reliability evaluation of non-reparable three-state systems. This shows the higher capability of the proposed methodology to assess large-scale system reliability in a reasonable computational time. Whereas, it is difficult or sometimes impossible to use both the previous methods in order to calculate system reliability of large-scale systems.

Insert Table 4 about here

6. Conclusion and future works

In this paper, we worked on non-repairable three-state systems and proposed a new method to calculate their reliabilities. To do this, a novel definition of states was first presented. Then, using a Markov process and the C-K theorem, the differential equations of the system were derived. Finally, the system reliability was obtained by solving the differential equations. The results of a comparison study on some non-repairable three-state sub-systems with different numbers of components showed that while the proposed method provided the same reliabilities compared to the UGF method and recursive algorithm, the computational time of the new method was much less than the ones of the two other procedures for larger sub-systems.

For future researches in this area, we recommend the followings:

- Calculating the reliability of other multi-state systems (systems with more than three states) through differential equations by a proper definition of the state
- Considering failure rates of the components as random variables
- Considering the failure rates of the component as fuzzy variables
- Considering the failure rates of the component time dependent
- Considering repairable components
- Considering the assumption that some failed components may be damaging the system (critical failures)

- ❖ Using the proposed method in redundancy allocation problems with the cost, weight, and the like as the constraints

Acknowledgment

The authors are thankful for constructive comments of anonymous reviewers. Taking care of the comments certainly improved the presentation of the manuscript. This research was financially supported by Iranian National Science Foundation under grant number 92020691, for which the authors are thankful.

References

Ding Y, Lisniaski A, Fuzzy universal generating function for multi-state system reliability assessment, *Fuzzy Sets and Systems* 159 (2008): 307–324.

Gnedenko B, Ushakov I, *Probabilistic reliability engineering*, Wiley, NY, Chichester, Brisbane, (1995).

Higashiyama Y, A factored reliability formula for weighted-k-out-of-n system, *Asia-Pacific Journal of Operations Research* 18 (2001): 61–66.

Kuo W, Wan R, Recent advances in optimal reliability allocation, *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems Humans* 37 (2007): 143–156.

Levitin G, Lisnianski A, Ben-Haim H, Elmakis D, Redundancy optimization for series-parallel multi-state systems, *IEEE Transactions on Reliability* 47 (1998): 165–172.

Levitin G, Lisnianski A, Joint redundancy and maintenance optimization for multi-state series-parallel systems, *Reliability Engineering and System Safety* 64 (1998): 33–42.

Levitin G, Lisnianski A, Structure optimization of power system with bridge topology. *Electric Power Systems Research* 45 (1998): 201–208.

Levitin G, Lisnianski A, Survivability maximization for vulnerable multi-state systems with bridge topology, *Reliability Engineering and System Safety*, 70 (2000): 125–140.

Levitin G, *Universal generating function and its applications*, Springer, Berlin, (2005).

Li W, Zuo M, Reliability evaluation of multi-state weighted k-out-of-n systems. *Reliability Engineering and System Safety* 93 (2008): 160–167.

Lisnianski A, Levitin G, Ben-Haim H, Structure optimization of multi-state system with time redundancy, *Reliability Engineering and System Safety* 67 (2000): 103–112.

Lisnianski A, Levitin G, *Multi-state system reliability assessment, optimization, applications*, World Scientific, Singapore (2003).

Lisnianski A, Levitin G, Ben-Haim H, Elmakis D, Power system structure optimization subject to reliability constraints, *Electric Power Systems Research* 39 (1996):145–152.

MATLAB Version 7.10.0.499 (R2010a), *The MathWorks*, Inc. Protected by U.S. and international patents.

Newell GF, *Applications of Queuing Theory*, 2nd ed; Chapman & Hall, London, (1982).

Ouzineb M, Nourelfath M, Gendreau M, Tabu search for the redundancy allocation problem of homogenous series–parallel multi-state systems, *Reliability Engineering and System Safety* 93 (2008): 1257–1272.

Tian Z, Levitin G, Zuo M, A joint reliability–redundancy optimization approach for multi-state series–parallel systems, *Reliability Engineering and System Safety*, 94 (2009): 1568–1576.

Ushakov I, The method of generalized generating sequences, *European Journal of Operational Resea*rch 125 (2000): 316–323.

Wu JS, Chen RJ, An algorithm for computing the reliability of a weighted-k-out-of-n system. *IEEE Transactions on Reliability* 43 (1994): 327–328.

Table 1: The matrix model of the example

n=3	(3,0)	(2,1)	(2,0)	(1,2)	(1,1)	(0,3)	(1,0)	(0,2)	(0,1)	(0,0)
(3,0)	0	$3\lambda_1$	$3\lambda_2$	0	0	0	0	0	0	0
(2,1)	0	0	λ_3	$2\lambda_1$	$2\lambda_2$	0	0	0	0	0
(2,0)	0	0	0	0	$2\lambda_1$	0	$2\lambda_2$	0	0	0
(1,2)	0	0	0	0	$2\lambda_3$	λ_1	0	λ_2	0	0
(1,1)	0	0	0	0	0	0	λ_3	λ_1	λ_2	0
(0,3)	0	0	0	0	0	0	0	$3\lambda_3$	0	0
(1,0)	0	0	0	0	0	0	0	0	λ_1	λ_2
(0,2)	0	0	0	0	0	0	0	0	$2\lambda_3$	0
(0,1)	0	0	0	0	0	0	0	0	0	λ_3
(0,0)	0	0	0	0	0	0	0	0	0	0

Table 2:The triple failure rates of components

I	λ_{i1}	λ_{i2}	λ_{i3}
1	0.008	0.004	0.006
2	0.006	0.003	0.005
3	0.009	0.0045	0.0055
4	0.009	0.005	0.007
5	0.005	0.002	0.004
6	0.007	0.002	0.004

Table 3: Comparison results

Number of components in a sub-system	**Proposed method**		**UGF method**		**Recursive method**	
	Reliability	Time	Reliability	Time	Reliability	Time
[1 1 1 1 1 1]	0.095292153001846	0.0656	0.095292153001846	0.0574	0.095292153001846	0.0427
[2 2 2 2 2 2]	0.496583514004698	0.0586	0.496583514004694	0.4629	0.496583514004696	0.1138
[3 3 3 3 3 3]	0.786476134630772	0.0611	0.786476134630760	87.0891	0.786476134630761	2.5351
[3 3 3 3 3 4]	0.793428887877881	0.0642	0.793428887877868	165.8123	0.793428887877866	3.5186
[3 3 3 3 4 4]	0.799589834224654	0.0654	0.799589834224633	301.1299	0.799589834224634	5.0299
[3 3 3 4 4 4]	0.839413937221604	0.0639	0.839413937221577	525.7848	0.839413937221577	7.0673
[3 3 4 4 4 4]	0.870312965681434	0.0652	0.870312965681405	911.0609	0.870312965681406	9.3240
[3 4 4 4 4 4]	0.886260344819907	0.0681	0.886260344819869	1512.6467	0.886260344819866	9.6901
[4 4 4 4 4 4]	0.915772045303859	0.0704	0.915772045303811	2559.4165	0.915772045303811	11.6725
[5 5 5 5 5 5]	0.967058717629245	0.0734	-	More than 4 hours	0.967058717629223	16.5213
[6 6 6 6 6 6]	0.987011788482230	1.0732	-	More than 4 hours	0.987011788482204	23.7146

Table 4: The results of employing the proposed method and the recursive algorithm in large-scale systems

Number of subsystems	Number of component in each subsystem	Proposed method		Recursive method	
		Reliability	Time (second)	Reliability	Time (second)
100	10	0.926465191004012	3.50	0.926465191004040	1505.8258
	20	0.999646546733508	15.95	0.999646546733502	3731.4684
500	10	0.639578721781569	10.49	0.639578721781592	6421.4562
	20	0.998491285747658	79.58	-	More than 4 hours
1000	10	0.346322178822173	12.63	-	More than 4 hours
	20	0.995734802698279	82.15	-	More than 4 hours

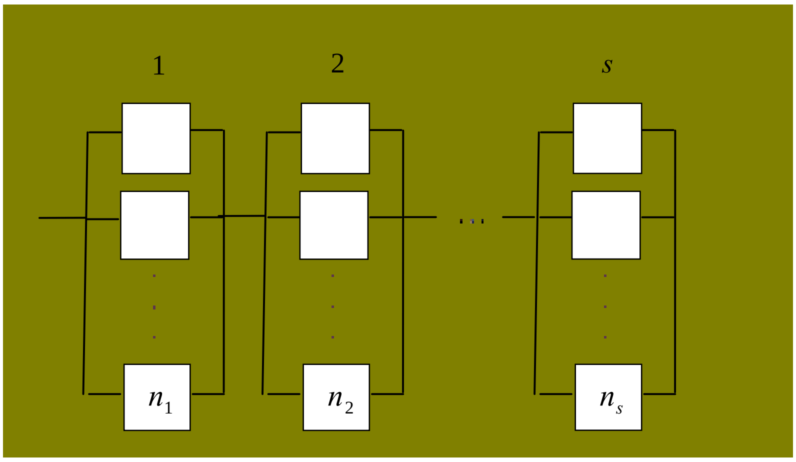

Fig.1: The system structure

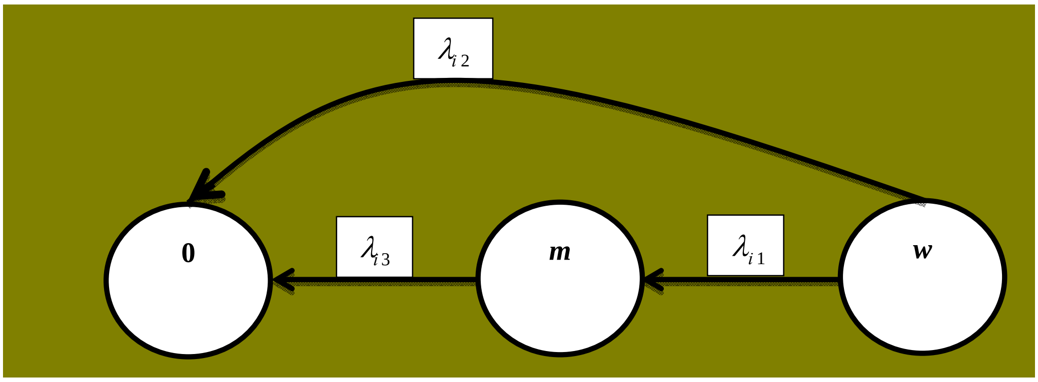

Fig. 2: State transition in a three-state component of the *i*th sub-system

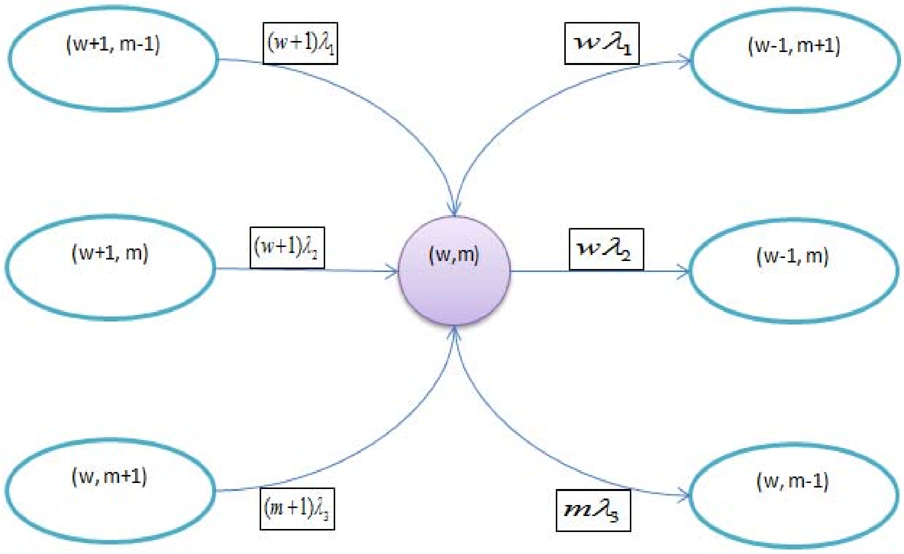

Fig. 3: State space diagram of a sub-system

Step 1 : determine number of component (n) and the failure rates (λ)
Step 2: obtain the state space
Step 3: obtain matrix model according to the state space
Step 4: obtain differential equations of all states
Step 5: solve differential equations and evaluate the reliability of the system

Fig. 4: The steps involved in the proposed method to obtain reliability

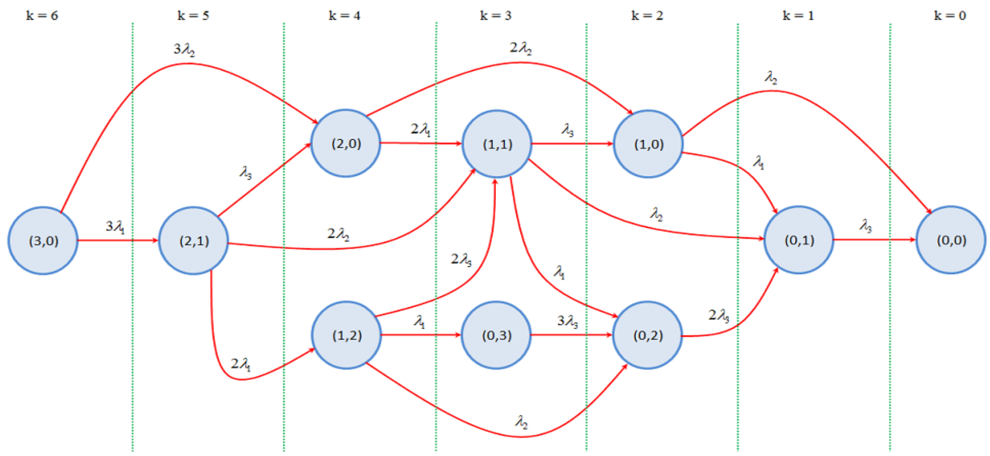

Fig. 5: Space state diagram of the example

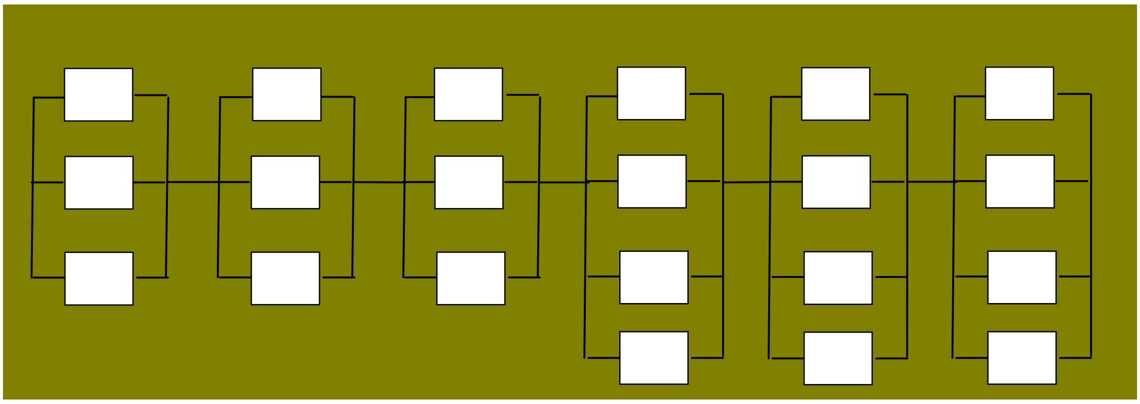

Fig. 6: System configuration of the example [3 3 3 4 4 4]

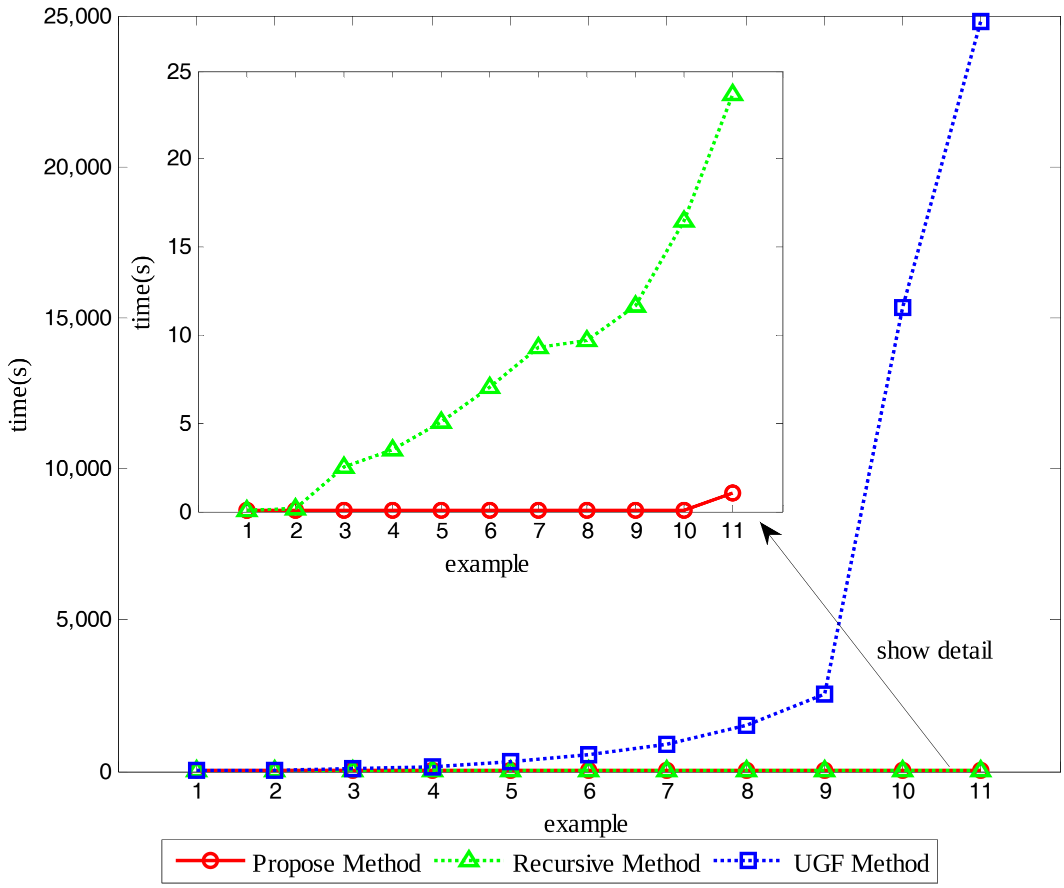

Fig. 7: CPU time comparisions of the three methods under investigation