Applied Mathematical Modelling 39 (2015) 5681–5692

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

Service reliability modeling of distributed computing systems with virus epidemics

Yan-Fu Li [a], Rui Peng [b,*]

[a] Ecole Centrale Paris – SUPELEC, Paris, France
[b] Dongling School of Economics and Management, University of Science & Technology Beijing, Beijing, China

ARTICLE INFO

Article history:
Received 21 November 2012
Received in revised form 17 December 2014
Accepted 12 January 2015
Available online 29 January 2015

Keywords:
Service reliability
Distributed computing system
Virus epidemics
Continuous-state model
Differential equations
Universal generating function

ABSTRACT

Distributed computing (DC) system is widely implemented due to its low setup cost and high computational capability. However, it might be vulnerable to malicious attacks like computer virus due to its network structure. The service reliability, defined as the probability of fulfilling a task before a specified time, is an important metric of the quality of a DC system. This paper attempts to model and compute the service reliability for the DC system under virus epidemics. Firstly, the DC system architecture is modeled by an undirected graph whose nodes (i.e. computers) have a continuous-state model representing its computational capability. Then a set of epidemic differential equations are formulated and solved to obtain the state dynamics of each node under the virus epidemics. A universal generating function (UGF) based approach is proposed to calculate the service reliability of DC system. Numerical results show the effectiveness of the proposed method. The sensitivity analysis on the model parameters, the comparison with centralized computing system and the optimization of defense level parameter are also conducted.

1. Introduction

Distributed computing (DC) system [1] is a collection of multiple autonomous computers that can communicate through a computer network to solve a large computational task. The purpose of the DC system is to coordinate the use of shared resources and provide communication services to the users [2]. Comparing to the centralized computing system, the DC system possesses many advantages, such as high performance, low setup cost, and potential for enhanced reliability [3–5]. Therefore, the DC system has gained increasing popularity in many application fields such as distributed software/hardware system [4,6], distributed power generation [7], distributed sensor system [8], and etc.

Like many other computing systems, the service quality of the DC system is of high concern to the majority of practitioners. Service reliability, which measures the capability of a system to accomplish its tasks on time, is a very important metric of DC system service quality [9]. Many research works have been devoted to modeling and analysis of the reliability (including service reliability) of DC systems [9–13]. However, most of the previous research works have focused on the failures caused by the 'unintentional' defects embedded in the DC hardware infrastructure and the installed software. In practice, external factors such as infective computer virus become widely spread in the current computer networks [14]. In this paper, we focus on the type of virus which can reproduce themselves and infect other computers in the network. If the virus

* Corresponding author. Tel.: +86 13051540519.
E-mail address: pengrui1988@ustb.edu.cn (R. Peng).

http://dx.doi.org/10.1016/j.apm.2015.01.036
0307-904X/

successfully parasitizes one computer, it will rapidly copy itself, consume the computing resources (e.g. CPU and memory) of the host, and attempt to infect other healthy computers via network connections (e.g. email, FTP transfer, message exchange, etc). This process will repeat on other infected computers and may eventually lead to a great loss of computational capability of the whole DC system if the situation is not attended to.

Protecting the DC system against the virus attacks becomes an increasingly important issue [6] and this type of protection is clearly different from the protection against 'acts of nature' or 'accidents' [15]. For example, the CPU breakdown in a computer usually will not affect the operations of other computers connected in the same network. In the literature of reliability research, many studies have been devoted to intentional attack protections by designing protection strategies for different systems (e.g. power substations, defense systems, etc) [16–20], but few have investigated the attacks with epidemic characteristics, such as computer virus [21]. In the field of epidemiology modeling, some research works have addressed the virus spreading issue in computer networks, but the emphasis is on the speed and range of the spreading [21–23]. To bridge the gap, in this work the service reliability of the DC system is modeled and computed under the virus epidemics with the consideration of possible system noises.

The rest of this paper is organized as follows. In Section 2, the general model of virus epidemics is proposed: the continuous-state model is used to describe the computational capability of each node in the DC system and the epidemic differential equations are set and solved to obtain the time-dependent state index. In Section 3, service reliability is defined based on the virus epidemic model and the universal generating function (UGF) technique is adopted for computing service reliability. Section 4 illustrates the proposed model on a numerical example with (1) the sensitivity analysis on the defense level parameter and the processing speed coefficient, (2) the comparison with centralized computing system and (3) the optimization of defense level parameter. Section 5 concludes this study with some possible future research directions.

2. Modeling of virus spreading in distributed computing systems

Notations

Ω	range of continuous state $\Omega = [0, 1]$, where 0 indicates the perfect functioning state and 1 indicates the complete failure state
T	system time
N	total number of nodes in the computer network
G	the undirected graph representing the computer network
V	the set of nodes in the computer network
v_i	node i in the computer network
L	the set of communication channels in the computer network
l_{ij}	the communication channel that links node v_i and v_j
$\mu_i(t)$	the state index of node i at time t
Ψ_i	the neighborhood set of node v_i
E_i	the set of subtasks distributed to node i
δ_i	defense parameter at node v_i
ξ_k	percentage of the raw data in sub-task k
d_k	the amount of data related to subtask k
$\theta_{ki}(t)$	data processing speed of subtask k by node v_i at time t
α_{ki}	processing speed coefficient which links the processing speed to the node state
$R_{ki}(T)$	probability that all the transmission and processing operations of subtask k assigned to node i can be finished by time t
K	total number of subtasks

In this work, the DC system is modeled as an undirected graph $G = (V, L)$, where $V = \{v_i | 1 \leqslant i \leqslant N\}$ is the set of computers (nodes), and $L = \{l_{ij} | 1 \leqslant i \leqslant N, i < j \leqslant N\}$ is the set of communication channels (links) connecting the nodes. It is noted that many authors have assumed homogeneous elements (no difference between nodes and links) in DC system [9,10]. However, for the virus epidemic modeling, nodes are usually treated as infectious components while the links are treated as the non-infectious channels for virus spreading [12,23].

2.1. A continuous-state reliability model of individual nodes

Markov chain is one of the conventional approaches for modeling DC system reliability with a number of discrete states [13], where each node has two states: online (functioning state) or offline (failure state) and the transition diagram is established to model the system state changes. However, the size of Markov state space grows exponentially with the increase of the number of nodes and the degradation states [24]. Moreover, when a node is under virus infection, it will not completely lose its computational capability in a short time. Once a virus successfully resides onto a node, it attaches itself to some executable files. Its code will be executed when one user attempts to launch an infected program. After the execution of

its code, the virus may replicate itself into other programs. Progressively, more and more programs will be infected and the running virus codes will consume greater amount of computing resources and slow down the entire computer. Therefore, a two-state model is not sufficient to describe the degradation phenomena of individual nodes.

A feasible alternative is the continuous-state model, which has been considered by many studies on computer virus epidemiology [21–23] where the state of each node takes real values to represent the degradation condition of the computational capability. In addition, continuous state models have already been considered by a number of reliability researchers, since many real-world systems and components exhibit the continuous type of degradation [25–28].

The continuous-state model defines a state space $\Omega = [0, 1]$ representing all possible intermediate real-valued levels between the two extremes: '0' is perfect functioning state and '1' is the complete failure state. In addition, $\mu_i(t)$ denotes the state index of node i at time t. In the next section, we will apply the epidemic functions to solve $\mu_i(t)$.

2.2. Virus epidemic model

In this section, we describe the epidemic differential equations for the virus spreading in our model. From the assumptions made in Section 2.1, if a node v_i is at healthy state, then $\mu_i(t) = 0$. The deviation from the normal state represents the level of damage to the node. The strength of interactions between node v_i and its neighbor v_j is defined by the weight w_{ij} attributed to the edge e_{ij}. In this study, the weight is set to be proportional to the speed of the communication channel between two nodes. For node v_i, its state is dependent on the cumulative impacts from all its neighbors $v_j \in \Psi_i$ each proportional to the connection strength w_{ij} with some time delay t_{ij}, and its own ability to defend virus infection. Taking all the above factors into consideration, the status change of node v_i can be represented using the following epidemic differential equation:

$$\frac{d}{dt}\mu_i(t) = \sum_{j\in\Psi_i} w_{ij}\mu_j(t - t_{ij}) - \mu_i(t)\delta_i, \tag{1}$$

where Ψ_i is the set of neighboring nodes that have direct connections to node i and δ_i is the time independent parameter that represents the ability of node v_i to defend virus infection. δ_i is an important parameter with physical meanings. For example, the anti-virus software such as McAfee usually offers different levels of protections ranging from basic to total protections. The more protections the anti-virus offers the higher price it has. Based on the single epidemic equation in (1), we establish the following equation system to model the combined effects of virus spreading and anti-virus mechanism in the entire DC system.

$$\begin{cases} \frac{d}{dt}\mu_1(t) = \sum_{j\in\Psi_1} w_{1j}\mu_1(t - t_{1j}) - \mu_1(t)\delta_1 \\ \vdots \\ \frac{d}{dt}\mu_i(t) = \sum_{j\in\Psi_i} w_{ij}\mu_j(t - t_{ij}) - \mu_i(t)\delta_i \\ \vdots \\ \frac{d}{dt}\mu_N(t) = \sum_{j\in\Psi_N} w_{Nj}\mu_N(t - t_{Nj}) - \mu_N(t)\delta_N \end{cases}. \tag{2}$$

There are many methods to solve such differential equation systems like (2). The Laplace transform $L\{\mu(t)\} = \int_0^\infty \mu(t)e^{-st}dt$ is an effective approach with the property that $L\{\mu'(t)\} = sL\{\mu(t)\} - \mu(0)$. After taking the Laplace transform on the both sides of Eq. (2), the transformed equation system can be solved via linear algebra given the initial conditions $\mu_i(0) = 1, \mu_j(0) = 0$ $(j = 1, \ldots, N, j \neq i)$. Then the inverse Laplace transform is applied on the obtained solutions to finally derive the solutions to (2). In Section 2.3, an illustrative example is presented to obtain the node state index.

2.3. An illustrative example

Fig. 1 shows a 5-node DC system with the node 1 infected by a certain type of virus. The edge weight and node virus defending parameters are assigned with some preset values, as shown in Table 1. The edge weight is determined by the speed of data transmission, e.g. megabyte per second (mbps), between two nodes, whereas the virus defense parameter can be determined by the protection level of the anti-virus software installed in each node.

The initial conditions imply that $\mu_1(0) = 1, \mu_2(0) = 0, \mu_3(0) = 0, \mu_4(0) = 0, \mu_5(0) = 0$. Since the size of the virus (about few Kbits) is often very small comparing to the bandwidth of the communication channel, the time delay caused by network traffic is assumed negligible. Based on the conditions mentioned above, the virus epidemics equations are written as follows.

$$\begin{cases} \frac{d}{dt}\mu_1(t) = 0.04\mu_2(t) + 0.09\mu_3(t) + 0.03\mu_5(t) - 0.2\mu_1(t) \\ \frac{d}{dt}\mu_2(t) = 0.04\mu_1(t) + 0.01\mu_3(t) - 0.05\mu_2(t) \\ \frac{d}{dt}\mu_3(t) = 0.09\mu_1(t) + 0.01\mu_2(t) + 0.06\mu_4(t) + 0.02\mu_5(t) - 0.2\mu_3(t) \\ \frac{d}{dt}\mu_4(t) = 0.06\mu_3(t) + 0.05\mu_5(t) - 0.125\mu_4(t) \\ \frac{d}{dt}\mu_5(t) = 0.03\mu_1(t) + 0.02\mu_3(t) + 0.05\mu_4(t) - 0.1\mu_5(t) \end{cases}. \tag{3}$$

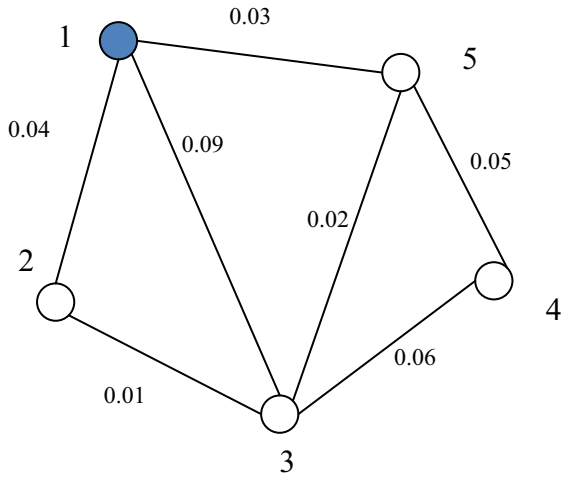

Fig. 1. A 5-node DC system with the node 1 infected.

Table 1
Weight of edges and virus defense parameter.

Edge	e_{12}	e_{13}	e_{15}	e_{23}	e_{34}	e_{35}	e_{45}
Weight	0.04	0.09	0.03	0.01	0.06	0.02	0.05
Node	1	2	3	4	5		
	0.2	0.05	0.2	0.125	0.1		

The eigenvalue method is used to solve (3). The solutions are as follows:

$$\begin{cases} \mu_1(t) = 0.4344e^{-0.304t} + 0.3185e^{-0.162t} + 0.0980e^{-0.146t} + 0.0015e^{-0.050t} + 0.1476e^{-0.013t} \\ \mu_2(t) = -0.0504e^{-0.304t} - 0.1267e^{-0.162t} - 0.0578e^{-0.146t} + 0.0310e^{-0.050t} + 0.2039e^{-0.013t} \\ \mu_3(t) = -0.4585e^{-0.304t} + 0.1485e^{-0.162t} + 0.1609e^{-0.146t} - 0.0056e^{-0.050t} + 0.1547e^{-0.013t} \\ \mu_4(t) = 0.1708e^{-0.304t} - 0.4046e^{-0.162t} + 0.0846e^{-0.146t} - 0.0157e^{-0.050t} + 0.1648e^{-0.013t} \\ \mu_5(t) = -0.0608e^{-0.304t} + 0.1236e^{-0.162t} - 0.2280e^{-0.146t} - 0.0170e^{-0.050t} + 0.1821e^{-0.013t} \end{cases}. \quad (4)$$

It can be derived from (4) that

$$\lim_{t\to\infty} \mu_i(t) = 0, \quad (5)$$

which implies that the DC system would end up in a totally healthy state in the long run. Such a conclusion can be attributed to the continuous efforts of virus defending mechanism. The plot of $\mu_i(t)$ over time shows the trend of the expected behaviors of each node.

As can be seen from Fig. 2, the state index of node 1 first drops steeply from 1 with time and then it begins to reduce more smoothly. This phenomenon is due to the virus defending mechanism. The curves of node 2, 3, 4 and 5 are similar. They all start from 0 and gradually increase. The node which has a larger weighted edge connecting with node 1 is expected to be infected faster than the others at the beginning. This situation is validated in the form of a steeper increasing curve of $\mu_2(t)$. It is also worth noting that the state index of each node becomes relatively stable as time increases.

In the next Section, the service reliability is modeled and computed.

3. Service reliability modeling of distributed system under virus epidemics

Based on the epidemic model of virus spreading presented in Section 2, this section derives the service reliability of the DC system. Suppose that the entire task is divided into K sub-tasks which are distributed to the N nodes for processing. One node can process multiple subtasks and a subtask can be distributed onto multiple nodes for processing. The service reliability is usually considered as the probability of successfully completing a target task within a predefined period of time [9,29]. In this work, we adopt this definition into the context of virus epidemic and regard the service as successful if it is finished within time T. Prior to the reliability model, the assumptions for the task processing and transmission are presented as follows:

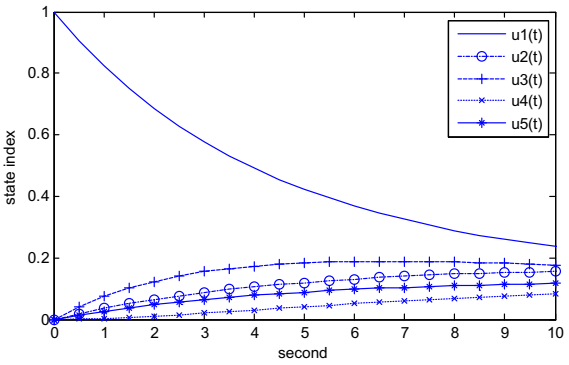

Fig. 2. Plot of $\mu_i(t)$ over time.

1. Each node can execute multiple subtasks simultaneously. A node starts to execute an assigned subtask immediately after it gets all necessary inputs, and the data processing speed depends on its state and the type of subtask.
2. Each link has a data transmission speed (bandwidth), of which a stable portion is occupied by one subtask throughout the entire task transmission and processing period. This assumption is in line with the multiplexing technique for computer communications.
3. The data transmission time within each node is negligible.
4. For each sub-task at one node, the data is transmitted through a same set of links before and after the execution.
5. The hardware failures due to natural causes are not considered because these failure probabilities are usually very low and the focus of our study is on virus spreading.

3.1. Service reliability of one subtask distributed to one node

Let d_k denote the amount of data related to sub-task k. It is noted that d_k contains two different sets of data: (1) the raw data forwarded to the processing node for processing; (2) the output data of the processing node which has to be sent back to the initiating node. Let ξ_k denote the percentage of the raw data in sub-task k, then $\xi_k \cdot d_k$ is the amount of raw data to be processed on certain node and $(1-\xi_k)\cdot d_k$ is the amount of result data to be returned to the initiating node. The data is transmitted between node v_i and the node that initiates the subtask, through a same set of links L_{ki} before and after the execution of the sub-task at one node. Thus the data transmission speed for sub-task k transmitted to resource v_i is

$$s_{ki} = \min_{l_j \in L_{ki}}(l_j), \tag{6}$$

where l_j is the transmission speed (or bandwidth) of the jth link in L_{ki} occupied by the sub-task k executed by node i. Hence the transmission time of the sub-task k is obtained as follows:

$$T_{ki} = \frac{d_k}{s_{ki}}. \tag{7}$$

Let E_i denote the set of subtasks distributed to node i. It can be obtained from (7) that it takes $t_{ki} = \xi_k T_{ki}$ units of time for node i to start processing subtask k. Similarly the time needed for node i to send back the results of subtask k can be obtained as $\tau_{ki} = (1-\xi_k)T_{ki}$.

The processing speed of subtask k by node v_i: $\theta_{ki}(t)$ is negatively related to the state index $\mu_i(t)$ of node i at time t . In practice, the relationship between $\theta_{ki}(t)$ and $\mu_i(t)$ can be estimated from real data as follows

$$\theta_{ki}(t) = f_{ki}(\mu_i(t)) + \varepsilon_{ki}(t), \tag{8}$$

where $f_{ki}(\cdot)$ is a decreasing function defined on $[0,1]$ and $\varepsilon_{ki}(t)$ is the noise of the processing speed following a Gaussian process with mean 0. To judge whether node i can finish processing subtask k before time t, we need to calculate the amount of data of subtask k that can be processed by node i from t_{ki} to $T-\tau_{ki}$, which can be expressed as

$$Y_{ki} = \int_{t_{ki}}^{T-\tau_{ki}} \theta_i(t)dt = \int_{t_{ki}}^{T-\tau_{ki}} [f_{ki}(\mu_i(t)) + \varepsilon_{ki}(t)]dt = \int_{t_{ki}}^{T-\tau_{ki}} f_{ki}(\mu_i(t))dt + \int_{t_{ki}}^{T-\tau_{ki}} \varepsilon_{ki}(t)dt. \tag{9}$$

The probability that node i can finish processing subtask k can be calculated as

$$R_{ki}(T) = \Pr(Y_{ki} > \xi_k d_k). \tag{10}$$

Denote $Z_{ki} = \int_{t_{ki}}^{T-\tau_{ki}} \varepsilon_{ki}(t)dt$. According to the property of Gaussian process, Z_{ki} itself is a normal random variable with mean 0. The variance of Z_{ki} and thus of Y_{ki} can be obtained as follows:

$$V(Y_{ki}) = V(Z_{ki}) = E(Z_{ki}^2) = \int_{t_{ki}}^{T-\tau_{ik}} \int_{t_{ki}}^{T-\tau_{ki}} E(\varepsilon_{ki}(t_1) \cdot \varepsilon_{ki}(t_2))dt_1 dt_2. \tag{11}$$

It can be seen that the variance of Y_{ki} is determined by the covariance function of the Gaussian process, which describes the covariance of noises at any pair of time points. Different covariance functions need to be used depending on the mechanism of the noises associated with the subtask processing speed. Specifically, if the covariance of the noises at different time points is 0 and the noise variance at any time t is a constant $\sigma_{ki}^2(t) = \sigma_{ki}^2$, we have

$$V(Y_{ki}) = \int_{t_{ki}}^{T-\tau_{ik}} \int_{t_{ki}}^{T-\tau_{ki}} E(\varepsilon_{ki}(t_1) \cdot \varepsilon_{ki}(t_2))dt_1 dt_2 = \int_{t_{ki}}^{T-\tau_{ik}} E(\varepsilon_{ki}(t_1) \cdot \varepsilon_{ki}(t_1))dt_1 = \sigma_{ki}^2(T - \tau_{ik} - t_{ki}). \tag{12}$$

Furthermore, we have

$$R_{ki}(T) = \Pr(Y_{ki} > \xi_k d_k) = 1 - \Phi\left(\frac{\xi_k d_k - \int_{t_{ki}}^{T-\tau_{ki}} f_{ki}(\mu_i(t))dt}{\sqrt{V(Y_{ki})}}\right), \tag{13}$$

where $\Phi(\cdot)$ represents the cumulative probability function of standard normal distribution.

3.2. Total service reliability

In this section, we utilize the UGF approach to derive the total service reliability based on the reliability index obtained in the section above for the individual subtask on one single node. UGF was first introduced by Ushakov in 1986 [30] and it proves to be very effective in evaluating reliability of complex multi-state systems. Much research has been done on incorporating UGF into reliability analysis of various k-out-of-n systems, series–parallel systems, weighted voting systems, acyclic information networks, and manufacturing systems [31–34]. The UGF of a discrete random value X is defined as a polynomial,

$$u(z) = \sum_{w=0}^{W-1} p_w z^{x_w}, \tag{14}$$

where the variable X has W possible values and p_w is the probability that X takes the value x_w.

In our case, the UGF of each subtask k distributed to node i is defined as

$$u_{ki}(z) = R_{ki} \cdot z^{\{k\}} + (1 - R_{ki}) \cdot z^{\phi}, \tag{15}$$

where ϕ denotes the empty set. The UGF representing two different subtasks distributed to node i can be obtained as

$$\begin{aligned} u_{ki}(z) \underset{union}{\otimes} u_{mi}(z) &= \sum_{h=0}^{1} p_h \cdot z^{S_h} \cdot \sum_{l=0}^{1} q_l \cdot z^{B_l} \\ &= \sum_{h=0}^{1}\sum_{l=0}^{1} p_h q_l \cdot z^{U(S_h, B_l)} \\ &= R_{ki}R_{mi}z^{\{k,m\}} + R_{ki}(1 - R_{mi})z^{\{k\}} + (1 - R_{ki})R_{mi}z^{\{m\}} + (1 - R_{ki})(1 - R_{mi})z^{\phi} \end{aligned}, \tag{16}$$

where S_k and B_l represent the sets of subtasks, and $\underset{union}{\otimes}$ is the proposed union composition operator. By iteratively combining the UGF representing all the subtasks distributed on node i, the UGF of the node i can be obtained as

$$U_i(z) = \sum_{h=0}^{H} P_h \cdot z^{Q_h}. \tag{17}$$

The UGF of the DC system can be obtained by combining the UGF of all the nodes together as

$$U(z) = u_1(z) \underset{union}{\otimes} u_2(z) \ldots \underset{union}{\otimes} u_N(z) = \sum_{j=0}^{J} P_j \cdot z^{Q_j}. \tag{18}$$

The service reliability can be obtained as

$$\phi(U(z)) = \phi\left(\sum_{j=0}^{J} P_j \cdot z^{Q_j}\right) = \sum_{j=0}^{J} P_j \cdot 1(Q_j = \{1, \ldots, K\}), \tag{19}$$

where $\phi(\cdot)$ is the redistributive operator designed to obtain the system service reliability.

4. A numerical example

To illustrate the proposed method for DC service reliability computation, the example in Section 2.3 is extended. Suppose that a task needs to be processed within T = 5.5 s. The resource management server (RSM) has divided the task into three

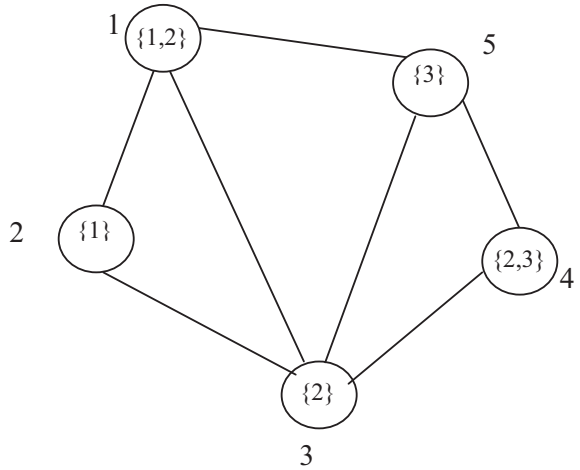

Fig. 3. The 5-node DC system with distributed subtasks.

Table 4
Bandwidths of the edges.

Edge	e_{12}	e_{13}	e_{15}	e_{23}	e_{34}	e_{35}	e_{45}
Bandwidth	40 mbps	90 mbps	30 mbps	10 mbps	60 mbps	20 mbps	50 mbps

subtasks and dispatched them as $E_1 = \{1,2\}$, $E_2 = \{1\}$, $E_3 = \{2\}$, $E_4 = \{2,3\}$, $E_5 = \{3\}$, as shown in Fig. 3. In real industrial practice, the algorithm for determining task dispatching is very complicated and interested readers are encouraged to refer to [1]. The size and the percentage of raw data for each subtask are given as $d_1 = 20$ mbits, $d_2 = 30$ mbits, $d_3 = 20$ mbits and $\xi_1 = \xi_2 = \xi_3 = 0.5$. For simplicity, let the relationship of processing speed and the node state have the following form:

$$\theta_{ki}(t) = \alpha_{ki}(1 - \mu_i(t)) + \varepsilon_{ki}(t),$$

where α_{ki} is the processing speed coefficient that links the processing speed to the node state. It is assumed that $\alpha_{ki} = 4$ and $E(\varepsilon_{ki}(t_1) \cdot \varepsilon_{ki}(t_2)) = 1(t_1 = t_2)$ for $k = 1,2,3$ and $i = 1,\ldots,5$, where 1(TRUE) = 1 and 1(FALSE) = 0. Subtasks 1 and 2 need to get their inputs from and send back the results to node 1. Subtask 3 needs to get inputs from and send back results to node 5.

The bandwidths of the edges are provided in the Table 4.

It is assumed that $L_{12} = \{e_{12}\}$, $L_{23} = \{e_{13}\}$, $L_{24} = \{e_{13}, e_{34}\}$, $L_{34} = \{e_{45}\}$. Note that e_{13} is in both L_{23} and L_{24}. In this study, we assume that 50% bandwidth of e_{13} is occupied by the transmission of sub-task 2 executed by node 3, and the other 50% bandwidth of e_{13} is occupied by the transmission of sub-task 2 executed by node 4.

4.1. Reliability calculation

According to (6), we have

$$s_{12} = 40 \text{ mbps}, \quad s_{23} = 45 \text{ mbps}, \quad s_{24} = 45 \text{ mbps}, \quad s_{34} = 50 \text{ mbps}.$$

Furthermore, from (7) we have

$$T_{12} = 0.5 \text{ s}, \quad T_{23} = 0.67 \text{ s}, \quad T_{24} = 0.67 \text{ s}, \quad T_{34} = 0.4 \text{ s}$$

Thus, it follows

$$t_{12} = \tau_{12} = 0.25 \text{ s}, \quad t_{23} = \tau_{23} = 0.33 \text{ s}, \quad t_{24} = \tau_{24} = 0.33 \text{ s}, \quad t_{34} = \tau_{34} = 0.2 \text{ s}.$$

According to (9), we have

$$E(Y_{11}) = 22 - 4\int_0^{5.5} \mu_1(t)dt = 8.0790, E(Y_{21}) = 22 - 4\int_0^{5.5} \mu_1(t)dt = 8.0790,$$

$$E(Y_{12}) = 20 - 4\int_{0.25}^{5.25} \mu_2(t)dt = 18.4662, E(Y_{23}) = 19.36 - 4\int_{0.33}^{5.17} \mu_3(t)dt = 16.7281,$$

$$E(Y_{24}) = 19.36 - 4\int_{0.33}^{5.17} \mu_4(t)dt = 18.9785, E(Y_{34}) = 20.4 - 4\int_{0.2}^{5.3} \mu_4(t)dt = 20.0227,$$

$$E(Y_{35}) = 22 - 4\int_0^{5.5} \mu_5(t)dt = 20.7666,$$

where

$$\int_0^{5.5} \mu_1(t)dt = 3.4802, \int_{0.25}^{5.25} \mu_2(t)dt = 0.3834, \int_{0.33}^{5.17} \mu_3(t) = 0.6513,$$

$$\int_{0.33}^{5.17} \mu_4(t)dt = 0.0887, \int_{0.2}^{5.3} \mu_4(t)dt = 0.0943, \int_0^{5.5} \mu_5(t)dt = 0.3084,$$

are obtained from (4).

From (12) and $E(\varepsilon_{ki}(t_1)\cdot\varepsilon_{ki}(t_2)) = 1(t_1 = t_2)$, we have

$$V(Y_{ki}) = (T - \tau_{ik} - t_{ki}).$$

From (13), we have

$$R_{11}(T) = \Pr(Y_{11} > 10) = 1 - \Phi\left(\frac{1.9210}{\sqrt{5.5}}\right) = 0.2064, R_{21}(T) = \Pr(Y_{21} > 15) = 1 - \Phi\left(\frac{6.9210}{\sqrt{5.5}}\right) = 0.0016,$$

$$R_{12}(T) = \Pr(Y_{12} > 10) = 1 - \Phi\left(\frac{-8.4662}{\sqrt{5}}\right) = 0.9999,$$

$$R_{23}(T) = \Pr(Y_{23} > 15) = 1 - \Phi\left(\frac{-1.7281}{\sqrt{4.83}}\right) = 0.7842,$$

$$R_{24}(T) = \Pr(Y_{24} > 15) = 1 - \Phi\left(\frac{-3.9785}{\sqrt{4.83}}\right) = 0.9649, R_{34}(T) = \Pr(Y_{34} > 10) = 1 - \Phi\left(\frac{-10.0227}{\sqrt{5.1}}\right) = 1.0000,$$

$$R_{35}(T) = \Pr(Y_{35} > 10) = 1 - \Phi\left(\frac{-10.7666}{\sqrt{5.5}}\right) = 1.0000.$$

(The probability of finishing subtask 1 is $R_1(T) = 1 - (1 - R_{11}(T))(1 - R_{12}(T)) = 0.9999$. The probability of finishing subtask 2 is $R_2(T) = 1 - (1 - R_{21}(T))(1 - R_{23}(T))(1 - R_{24}(T)) = 0.9924$. The probability of finishing subtask 3 is $R_3(T) = 1 - (1 - R_{34}(T))(1 - R_{35}(T)) = 1.0000$. Thus $R(T) = 0.9999 * 0.9924 * 1 = 0.9924$. The complete UGF procedures are listed below. The UGF result includes more information than just a reliability value.)

From (15), we have

$$u_{11}(z) = 0.2064z^{\{1\}} + 0.7936z^{\phi}, u_{21}(z) = 0.0016z^{\{2\}} + 0.9984z^{\phi}, u_{12}(z) = 0.9999z^{\{1\}} + 0.0001z^{\phi},$$
$$u_{23}(z) = 0.7842z^{\{2\}} + 0.2158z^{\phi}, u_{24}(z) = 0.9649z^{\{2\}} + 0.0351z^{\phi}, u_{34}(z) = z^{\{3\}}, u_{35}(z) = z^{\{3\}}.$$

From (16), we have

$$U_1(z) = 0.0003z^{\{1,2\}} + 0.2061z^{\{1\}} + 0.0013z^{\{2\}} + 0.7933z^{\phi},$$

$$U_2(z) = 0.9999z^{\{1\}} + 0.0001z^{\phi},$$

$$U_3(z) = 0.7842z^{\{2\}} + 0.2158z^{\phi},$$

$$U_4(z) = 0.9649z^{\{2,3\}} + 0.0351z^{\{3\}},$$

$$U_5(z) = z^{\{3\}}.$$

Furthermore, we have

$$U_1(z) \underset{union}{\otimes} U_2(z) = 0.0016z^{\{1,2\}} + 0.9983z^{\{1\}} + 0.0001z^{\phi},$$

$$U_1(z) \underset{union}{\otimes} U_2(z) \underset{union}{\otimes} U_3(z) = 0.7845z^{\{1,2\}} + 0.2154z^{\{1\}} + 0.0001z^{\{2\}},$$

$$U_1(z) \underset{union}{\otimes} U_2(z) \underset{union}{\otimes} U_3(z) \underset{union}{\otimes} U_4(z) = 0.9923z^{\{1,2,3\}} + 0.0076z^{\{1,3\}} + 0.0001z^{\{2,3\}},$$

$$U(z) = U_1(z) \underset{union}{\otimes} U_2(z) \underset{union}{\otimes} U_3(z) \underset{union}{\otimes} U_4(z) \underset{union}{\otimes} U_5(z) = 0.9923z^{\{1,2,3\}} + 0.0076z^{\{1,3\}} + 0.0001z^{\{2,3\}},$$

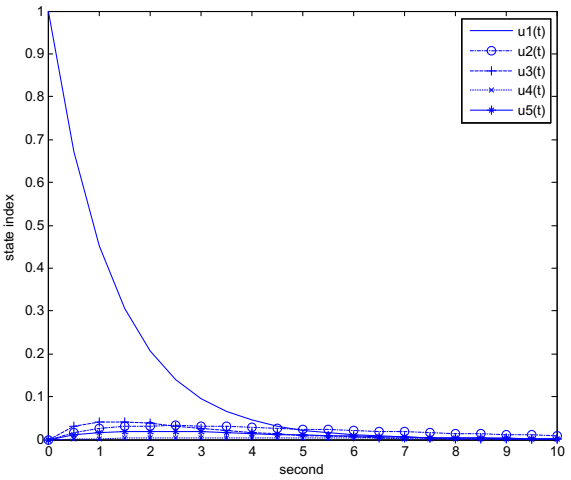

Fig. 4a. Plot of $\mu_i(t)$ over time with the defense parameters being 4 times larger than those in Table 1.

According to (19), the service reliability is 0.9923.

4.2. Sensitivity analysis

In this Section, the sensitivity analysis is performed on two model parameters, namely the defense level and the processing speed coefficient, that are influential to the computation of DC system service reliability.

In order to investigate the effects of defense level on the node states, Figs. 4a and 4b shows the expected node states when the defense parameter of each node is as four times/twice bigger as the defense parameter shown in Table 1. With comparison to the curves in Fig. 2, it clearly reflects the fact that the nodes are generally much healthier and the infected nodes return to healthy states more rapidly when stronger defense is available.

Fig. 5 shows the curves of service reliability as a function of processing speed coefficient for different levels of defense. It is assumed that the processing speed coefficient α_{ki} is the same for every k and i. It can be seen that the service reliability has a S-shaped increase when the processing speed coefficient increases and the higher service reliability is associated with the higher defense level. The S-shaped curve is attributed to the reason that the pdf of a normal distribution is higher nearer the center so that the reliability of a subtask distributed to a node increases faster when the absolute difference between Y_{ki} and $\xi_k d_k$ in (13) is smaller.

4.3. Comparison with centralized computing system

Consider the situation that there is only node 1 in the computing system and all the subtasks are assigned to it. As the cost for constructing and defending other nodes are saved, the defense parameter for node 1 has changed from $\delta_1 = 0.2$ to $\delta_1^{\sim}$. The initial state of node 1 is $\mu_1(0) = 1$. The state equation of node 1 can be expressed as

$$\frac{d}{dt}\mu_1(t) = -\delta_1^{\sim}\mu_1(t).$$

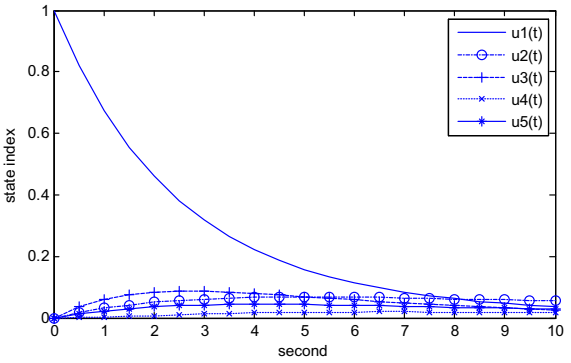

Fig. 4b. Plot of $\mu_i(t)$ over time with the defense parameters being twice larger than those in Table 1.

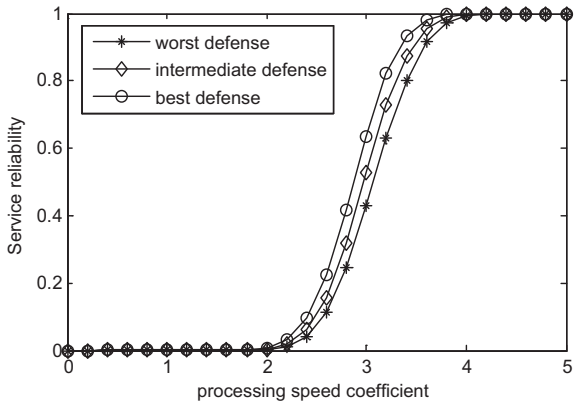

Fig. 5. Service reliability versus processing speed coefficient for different levels of defense.

It is easy to know that $\mu_1(t) = e^{-\tilde{\delta_1}t}$. For a fair comparison, it is still assumed that $\theta_{ki}(t) = 4(1-\mu_i(t)) + \varepsilon_{ki}(t)$ and $E(\varepsilon_{k1}(t_1)\cdot\varepsilon_{k1}(t_2)) = 1(t_1 = t_2)$ for k = 1,2,3.

Similarly, we can obtain

$$E(Y_{11}) = E(Y_{21}) = E(Y_{31}) = 22 - 4\int_0^{5.5}\mu_1(t)dt = 22 - \frac{4(1-e^{-\tilde{\delta_1}5.5})}{\tilde{\delta_1}},$$

$$V(Y_{k1}) = 5.5.$$

Furthermore we have

$$R_{11}(T) = \Pr(Y_{11} > 10) = 1 - \Phi\left(\frac{\frac{4(1-e^{-\tilde{\delta_1}T})}{\tilde{\delta_1}} - 12}{\sqrt{5.5}}\right),$$

$$R_{21}(T) = \Pr(Y_{21} > 15) = 1 - \Phi\left(\frac{\frac{4(1-e^{-\tilde{\delta_1}T})}{\tilde{\delta_1}} - 7}{\sqrt{5.5}}\right),$$

$$R_{31}(T) = \Pr(Y_{31} > 10) = 1 - \Phi\left(\frac{\frac{4(1-e^{-\tilde{\delta_1}T})}{\tilde{\delta_1}} - 12}{\sqrt{5.5}}\right).$$

Thus the reliability of the centralized system is better if

$$R_1(T) = R_{11}(T)R_{21}(T)R_{31}(T) > 0.9923.$$

when T = 5.5, the reliability of the centralized system is as shown in Fig. 6. The system reliability is greater than 0.9923 when the defense parameter is greater than 3.05.

Note that our comparison between centralized computing system and distributed computing system is based on the assumption that the infected node is curable regardless of the severity of the infection. In some other cases, the defense may become invalid when the infected node is overly damaged, i.e., its state index is above some threshold. In such cases, it is more reasonable to use distributed system structure. If some nodes in a distributed computing system have already become incurable, the defense mechanism can still prevent other nodes from infection and they may be able to complete the system mission. However, for the centralized computing system, it fails as long as the single node has become incurable.

4.4. Optimal allocation of defense level

In case the defense level of one individual node can be controlled by the amount of defense resource allocated to the node, the optimal allocation of defense level can be studied. For a DC system with N nodes, the optimal defense level allocation problem which maximizes the system reliability subject to a given cost can be formulated as

$$\text{Maximize} \quad R(\delta_1,\ldots,\delta_N),$$

$$\text{Subject to} \quad C(\delta_1,\ldots,\delta_N) \leqslant C_0,$$

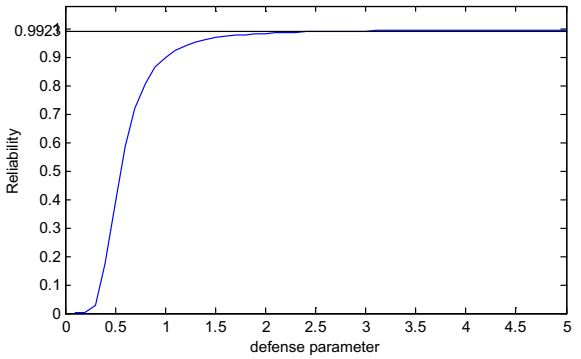

Fig. 6. Reliability of centralized system as a function of the defense parameter.

where $R(\delta_1,\ldots,\delta_N)$ and $C(\delta_1,\ldots,\delta_N)$ are respectively the system reliability and the total cost when the defense levels of the nodes are $\delta_1,\ldots,\delta_N$, and C_0 is the maximum allowable cost. For illustration, consider the optimal allocation of defense level for the system presented in part A of this section. Without loss of generality, it is assumed that

$$C(\delta_1,\ldots,\delta_5)=\delta_1+\delta_2+\delta_3+\delta_4+\delta_5,$$

and $C_0=0.2+0.05+0.2+0.125+0.1=0.575$.

The optimal defense levels of the nodes are found to be $(\delta_1,\ldots,\delta_5)=(0.575,0,0,0,0)$. The corresponding system reliability is 0.9968.

Note that our model is readily adapted to a more general case where the defense parameters take stepwise values instead of constant ones throughout the system mission time. To fulfill this, we need to divide the system mission time into certain periods such that the defense parameters are constant within each but can change in different periods. In this case, we need to first solve Eq. (2) to obtain the node state index in the first period given the initial condition at mission time 0. Then, we can solve (2) for the node state index in the second period given the node state index value at the end of the first period as the initial condition. Iteratively, the node state index throughout the whole system mission can be solved as a piecewise function of time. With the state indices of all the nodes, the system reliability can be calculated in similar procedures. The optimal defense resource allocation will be formulated as

$$\text{Maximize}\quad R((\delta_{11},\ldots,\delta_{1M}),\ldots,(\delta_{N1},\ldots,\delta_{NM})),$$

$$\text{Subject to}\quad C((\delta_{11},\ldots,\delta_{1M}),\ldots,(\delta_{N1},\ldots,\delta_{NM}))\leqslant C_0,$$

where M is the number of periods, $R((\delta_{11},\ldots,\delta_{1M}),\ldots,(\delta_{N1},\ldots,\delta_{NM}))$ and $C((\delta_{11},\ldots,\delta_{1M}),\ldots,(\delta_{N1},\ldots,\delta_{NM}))$ are respectively the system reliability and the total cost when the defense levels of the nodes are $\delta_{1k},\ldots,\delta_{Nk}$ for each period $1\leqslant k\leqslant M$, and C_0 is the maximum allowable cost.

5. Conclusions and future works

DC system is popular in industry because of its low setup and maintenance cost as well as high computational capability. However, due to the network nature of DC system, it might be vulnerable to virus attacks. This paper focuses on the computation of the service reliability of DC system under virus epidemics. The computational capability of individual node is modeled by a continuous-state model. The network topology of DC system is explicitly modeled as an undirected graph. A set of differential equations are formulated to describe the node state dynamics due to virus spreading. A universal generating function based approach is proposed to calculate the service reliability of the DC system. A numerical example is presented for illustration. The sensitivity analysis on the model parameters, the comparison with centralized computing system and the optimization of defense level parameter are also conducted.

The results show that enhancing the virus defense of each node i is an effective way of recovering the system from virus attacks, but it may be costly if every node has to be attended to. In future research, we will attempt to implement the developed methods onto larger scale DC systems. In addition, it is an essential issue to evaluate the risk in different fields [35–37]. Thus, another direction is to conduct risk analysis on our continuous state epidemic model.

Acknowledgement

The research report here is partially supported by the NSFC under Grant numbers 71231001, 71301009 and 71420107023, the Fundamental Research Funds for the Central Universities of China, FRF-TP-14-051A2, and by the MOE PhD supervisor fund, 20120006110025.

References

[1] G.R. Andrews, Foundations of Multithreaded, Parallel, and Distributed Programming, Addison-Wesley, 2000.
[2] S. Ghosh, Distributed Systems–An Algorithmic Approach, Chapman & Hall/CRC3, 2007.
[3] S. Kounev, Performance modeling and evaluation of distributed component-based systems using Queueing Petri Nets, IEEE Trans. Software Eng. 32 (7) (2006) 486–502.
[4] C.D. Lai, M. Xie, K.L. Poh, Y.S. Dai, P. Yang, A model for availability analysis of distributed software/hardware systems, Inf. Softw. Technol. 44 (2002) 343–350.
[5] N. Stankovic, K. Zhang, A distributed parallel programming framework, IEEE Trans. Software Eng. 28 (5) (2002) 478–493.
[6] M.J. Rutherford, A. Carzaniga, A.L. Wolf, Evaluating test suites and adequacy criteria using simulation-based models of distributed systems, IEEE Trans. Software Eng. 34 (4) (2008) 452–470.
[7] L.F. Wang, C. Singh, Reliability-constrained optimum placement of reclosers and distributed generators in distribution networks using an ant colony system algorithm, IEEE Trans. Syst. Man Cybern. Part C-Appl. Rev. 38 (6) (2008) 757–764.
[8] A. Kerrouche, J. Leighton, W.J.O. Boyle, Y.M. Gebremichael, T. Sun, K.T.V. Grattan, B. Taljsten, Strain measurement on a rail bridge loaded to failure using a fiber bragg grating-based distributed sensor system, IEEE Sens. J. 8 (11–12) (2008) 2059–2065.
[9] Y.S. Dai, M. Xie, K.L. Poh, G.Q. Liu, A study of service reliability and availability for distributed systems, Reliab. Eng. Syst. Safety 79 (2003) 103–112.
[10] H.M.A. Fahmy, Reliability evaluation in distributed computing environments using the AHP, Comput. Netw. 36 (2001) 597–615.
[11] M.S. Lin, M.S. Chang, D.J. Chen, Efficient algorithms for reliability analysis of distributed computing systems, Inf. Sci. 117 (1–2) (1999) 89–106.
[12] C. Lin, X. Jiang, H. Yin, Y.Z. Wang, Y.D. Hu, B.B. Xiong, Optimizing availability and QoS of heterogeneous distributed system based on residual lifetime in uncertain environment, J. Supercomputing 48 (3) (2009) 243–263.
[13] M. Xie, Y.S. Dai, K.L. Poh, C.D. Lai, Optimal number of hosts in a distributed system based on cost criteria, Int. J. Syst. Sci. 35 (6) (2004) 343–353.
[14] D.H. Shih, H.S. Chiang, C.D. Yen, Classification methods in the detection of new malicious emails, Inf. Sci. 172 (1–2) (2005) 241–261.
[15] V.M. Bier, A. Nagaraj, V. Abhichandani, Protection of simple series and parallel systems with components of different values, Reliab. Eng. Syst. Safety 87 (3) (2005) 315–323.
[16] G. Levitin, Optimal defense strategy against intentional attacks, IEEE Trans. Reliab. 56 (1) (2007) 148–157.
[17] K. Hausken, Strategic defense and attack for series and parallel reliability systems, Eur. J. Oper. Res. 186 (2) (2008) 856–881.
[18] G. Levitin, H. Ben-Haim, Importance of protections against intentional attacks, Reliab. Eng. Syst. Safety 93 (4) (2008) 639–646.
[19] G. Levitin, K. Hausken, Protection vs. redundancy in homogeneous parallel systems, Reliab. Eng. Syst. Safety 93 (10) (2008) 1444–1451.
[20] D. Shah, S. Zhong, Two methods for privacy preserving data mining with malicious participants, Inf. Sci. 177 (23) (2007) 5468–5483.
[21] J.C. Wierman, D.J. Marchette, Modeling computer virus prevalence with a susceptible-infected-susceptible model with reintroduction, Comput. Stat. Data Anal. 45 (1) (2004) 3–23.
[22] L. Buzna, K. Peters, D. Helbing, Modelling the dynamics of disaster spreading in networks, Physica A: Stat. Mech. Appl. (2006) 132–140
[23] W. Murray, The application of epidemiology to computer viruses, Comput. Secur. 7 (1988) 139–150.
[24] H. Pham, Software reliability and cost models: perspectives, comparison, and practice, Eur. J. Oper. Res. 149 (3) (2003) 475–489.
[25] L.A. Baxter, Continuum structures I, J. Appl. Probab. 21 (1984) 802–815.
[26] R.D. Brunelle, K.C. Kapur, Review and classification of reliability measures for multistate and continuum models, IIE Trans. 31 (12) (1999) 1171–1180.
[27] P.X. Liu, M.J. Zuo, Q.H. Meng, Using neural network function approximation for optimal design of continuous-state parallel–series systems, Comput. Oper. Res. 30 (3) (2003) 339–352.
[28] Q. Long, M. Xie, S.H. Ng, G. Levitin, Reliability analysis and optimization of weighted voting systems with continuous states input, Eur. J. Oper. Res. 191 (1) (2008) 238–250.
[29] Y.S. Dai, M. Xie, K.L. Poh, Availability modeling and cost optimization for the grid resource management system, IEEE Trans. Syst. Man Cybern. Part A-Syst. Humans 38 (1) (2008) 170–179.
[30] I. Ushakov, Universal generating function, Soviet J. Comput. Syst. Sci. 24 (5) (1986) 118–129.
[31] Y. Ding, M.J. Zuo, A. Lisnianski, W. Li, A framework for reliability approximation of multi-state weighted k-out-of-n systems, IEEE Trans. Reliab. 59 (2) (2010) 297–308.
[32] C.Y. Li, X. Chen, X.S. Yi, J.Y. Tao, Heterogeneous redundancy optimization for multi-state series-parallel systems subject to common cause failures, Reliab. Eng. Syst. Safety 95 (3) (2010) 202–207.
[33] R. Peng, M. Xie, S.H. Ng, G. Levitin, Element maintenance and allocation for linear consecutively connected systems, IIE Trans. 40 (11) (2012) 964–973.
[34] Y.F. Li, Y. Ding, E. Zio, Random fuzzy extension of the universal generating function approach for the reliability assessment of multi-state systems under aleatory and epistemic uncertainties, IEEE Trans. Reliab. 63 (1) (2014) 13–25.
[35] D.S. Wu, S.H. Chen, D. Olson, Business intelligence in risk management: some recent progresses, Inf. Sci. 256 (1-7) (2014) 1–7.
[36] D.S. Wu, D. Olson, Computational simulation and risk analysis: an introduction of state of the art research, Math. Comput. Modell. 58 (9) (2013) 1581–1587.
[37] D.S. Wu, S.C. Fang, D. Olson, J. Birge, Introduction to the special issue on optimizing risk management in services, Optimization 61 (10) (2012) 1175–1177.