

1 **System reliability analysis for independent and nonidentical components**
 2 **based on survival signature**

3 *Yide Zheng^a, Yi Zhang^{a*}, Jiarui Lin^a*

4 ^a Department of Civil Engineering, Tsinghua University, China

5 E-mail: zhang-yi@tsinghua.edu.cn

6 **Abstract**

7 Survival signatures have been widely used for analyzing arbitrary dependent system structures
 8 having failure events. It is challenging to study system reliability when diverse components are
 9 present since the fundamental premise of the survival signatures is that components should be
 10 interchangeable within the same subsystem. In this research, two methods based on the fundamental
 11 survival signature idea are suggested to examine the reliability of complex systems with independent
 12 but not necessarily identically distributed (INID) components. The first algorithm is based on the
 13 weighted random sampling (*WRS*) method to calculate the survival signatures. The second
 14 algorithm adopts the idea of divide-and-conquer in computing the probability structure. The
 15 application of these two algorithms and the exhaustive algorithm in analyzing the example systems
 16 are presented. The results show that the two algorithms can significantly reduce the computation
 17 time compared to the traditional methods. Finally, the developed algorithms are used in the
 18 reliability analysis of a real practical problem, the Tsinghua campus water supply pipeline system.
 19 The results regarding the predictions of system reliability as well as the component importance
 20 index are discussed.

21 **Keywords:** civil engineering systems and structures, reliability modeling and system
 22 optimization, divide-and-conquer, weighted sampling, independent but nonidentical distributed

24 **1. Introduction**

25 System reliability analysis is of great significance in engineering projects, especially for large
 26 complex systems including multi-state systems, network systems, and common cause failure
 27 systems [1], [2]. Many system analysis approaches have been developed for different types of
 28 systems, such as the fault tree (FT) model [3], the binary decision diagram (BDD) [4]-[7], and
 29 the Bayesian networks (BNs) [8],[9]. These methods can effectively split the system and
 30 identify the failure modes. However, from the clear and brief structure of the failure mode to
 31 the system failure probability, it is still a problem worth further study. For example, Kvassay
 32 [10] applied the method to obtain the BDD structure for systems, but for complex systems, such
 33 as systems with independent and different distributed components, the cost of calculating the
 34 system failure probability from the BDD structure is huge. Therefore, new algorithms are
 35 needed to simplify the process of calculating the failure probability. Coolen and Coolen-Maturi
 36 developed the survival signature approach [11], which divides the system reliability into system
 37 structure and probability structure. It enables full separation of the system structure from
 38 component probabilistic failure time and the only assumption is that the components are
 39 exchangeable when they are in the same type. The survival signature cannot fully exploit its
 40 benefits and simplify the system reliability for systems with independent and nonidentical
 41 distribution components since it does not satisfy the exchangeability assumption. The system

42 reliability of complex systems may be solved in a novel method by further generalizing the
 43 survival signature theory. The basics of survival signature theory are briefly discussed in the
 44 paragraphs that follow.

45 Suppose a system consists of m components with K types of components. Each type has
 46 m_k components and $\sum_{k=1}^K m_k = m$. The state vector $\underline{x} = (x_1, x_2, \dots, x_m) \in \{0,1\}^m$ is defined
 47 to describe the working state for all components while $x_i = 1$ if the i th component functions
 48 and $x_i = 0$ if not. $\phi = \phi(\underline{x}) : \{0,1\}^m \rightarrow \{0,1\}$ defines the system structure function while
 49 $\phi(\underline{x}) = 1$ if the system functions and $\phi(\underline{x}) = 0$ if not. For such a system, the survival
 50 signature is denoted by $\Phi(l_1, l_2, \dots, l_K)$ with $l_k = 0, 1, 2, \dots, m_k$ for $k = 1, 2, \dots, K$, which is
 51 defined to be the probability that the system functions given that l_k of its m_k components of
 52 type k work, for each $k \in \{1, 2, \dots, K\}$. There are $\binom{m_k}{l_k} = \frac{m_k!}{(m_k - l_k)! \cdot l_k!}$ state vectors $\underline{x}^k =$
 53 $(x_1^k, x_2^k, \dots, x_{m_k}^k)$ with l_k components working in the type k . Let S_{l_1, l_2, \dots, l_K} denote the set of all
 54 state vectors for the whole system when l_k components working in the type k ($k = 1, 2, \dots, K$)
 55 subsystem. When the components are exchangeable in the same subsystem, which means the
 56 probability of each state vector \underline{x} in S_{l_1, l_2, \dots, l_K} is the same, then the survival signature can be
 57 calculated by the formula below

$$\Phi(l_1, l_2, \dots, l_K) = \left[\prod_{k=1}^K \binom{m_k}{l_k}^{-1} \right] \times \sum_{\underline{x} \in S_{l_1, l_2, \dots, l_K}} \phi(\underline{x}). \quad (1)$$

58 Assume all the components are fully independent and components in the same subsystem
 59 are identically distributed, which is one of the conditions that satisfy the exchangeability
 60 assumption. Let $D_k(t) \in \{0, 1, 2, \dots, m_k\}$ denote the number of k type components working at
 61 time t and the components in the same type subsystem have a known cumulative distribution
 62 function (CDF), $F_k(t)$, then:

$$P(\bigcap_{k=1}^K \{D_k(t) = l_k\}) = \prod_{k=1}^K P(D_k(t) = l_k) = \prod_{k=1}^K \binom{m_k}{l_k} [F_k(t)]^{m_k - l_k} [1 - F_k(t)]^{l_k}. \quad (2)$$

63 Hence, the survival function of the system with K types of components can be formulated
 64 as:

$$P(T_s > t) = \sum_{l_1=0}^{m_1} \dots \sum_{l_K=0}^{m_K} \Phi(l_1, l_2, \dots, l_K) P(\bigcap_{k=1}^K \{D_k(t) = l_k\}). \quad (3)$$

65 In Eq. (3), the survival signature $\Phi(l_1, l_2, \dots, l_K)$ is also called system structure while
 66 $P(\bigcap_{k=1}^K \{D_k(t) = l_k\})$ is the probability structure in the survival function. The system structure
 67 is completely determined by the composition of the system and the system failure mode. While
 68 the probability structure describes the probability of the occurrence of different state vectors.
 69 The survival signature theory is used in many typical system reliability analyses with its concise
 70 computational form.

71 Studies on survival signature are mainly divided into two categories: 1. reliability analysis
 72 of different types of systems based on survival signature theory; 2. research on efficient
 73 computational algorithms of system structure. Aslett et.al [12] used survival signature in
 74 Bayesian inference to analyze network reliability. Feng et.al [13] analyzed the imprecise system
 75 reliability and the measure of component importance based on the survival signature and
 76 obtained the upper and lower bounds for system reliability. George et.al [14] extended the
 77 survival signature to non-repairable dependent failure problems and Huang [15] proposed a
 78 heuristic survival-signature-based approach for reliability-redundancy allocation. Samaniego
 79 and Navarro [16] considered comparing systems with heterogeneous components using the
 80 survival signature. Mi et.al [17] and Geng [18] applied it to solve common cause failure

81 problems. Coolen-Maturi [19] solved the problems in coherent systems with shared
 82 components on the concept of joint survival signature. Patelli et.al [20] proposed a simulation-
 83 based method to calculate it analytically. There are also many practical applications of the
 84 survival signature, such as systems with multi-state components [21],[22], the hydraulic system
 85 of wind turbines [23], reliability analysis of stress-strength models [24], coherent systems [25],
 86 and more studies can be found in [26]-[28]. In the second type of research, survival signature
 87 computational algorithms based on the graph structure [29],[30] and the binary decision
 88 diagram (BDD) structure [31]-[33] were proposed, and corresponding calculation methods
 89 have been applied for multi-state consecutive-k systems [34].

90 Despite advances in theory and the development of numerous applications, the assumption
 91 of exchangeability makes it difficult to analyze heterogeneous component systems. When the
 92 system with m components is divided into K groups and $K \ll m$, the application of the survival
 93 signature is convenient and concise. However, almost no two components are exactly the same
 94 in real engineering projects. There are $\prod_{k=1}^K (m_k + 1)$ cases when calculating $\Phi(l_1, l_2, \dots, l_K)$
 95 and one of the advantages of survival signature is that most $\Phi(l_1, l_2, \dots, l_K)$ are 0. But with the
 96 increase of K , especially when K approaches m or $K = m$, computing the system structure will
 97 be a huge computational challenge. Moreover, for some traditional system reliability solving
 98 methods, such as the fault tree and BDD, when there is a large amount of state vectors \underline{x} that
 99 lead to the system function $\phi(\underline{x}) = 1$, it will take too much time for system analysis.

100 Therefore, in this paper, systems with independent but not necessarily identically
 101 distributed (INID) components are explored. The components no longer adhere to the premise
 102 of interchangeability, but to increase computing efficiency, the concept of system reliability
 103 into system structure and probability structure is adopted. In order to address two problems in
 104 system structure and probability structure that emerge when the assumptions are reduced to the
 105 point where the components are only independently distributed but not necessarily so, a
 106 systematic approach is developed. The structure of the essay is as follows. Based on the survival
 107 signature theory, Section 2 derives the survival function for distributed systems with just
 108 components that are independently distributed but not identical. The generic probability
 109 structure function is developed in Section 3. Using the use of a recently created divide-and-
 110 conquer algorithm, the value of the probability structure function is calculated. In Section 4,
 111 the developed algorithm for computing the system structure in the survival signature is
 112 presented. In Section 5, the two algorithms are combined to form a complete approach to
 113 computing the importance index for large-scale systems. To demonstrate the proposed method,
 114 a pipeline system from the Tsinghua campus is investigated in Section 6. Finally, conclusions
 115 and discussions on possible future developments are provided in Section 7.

2. System with independent and nonidentical components

2.1 New form for system structure

116 As mentioned above, the traditional survival signature theory requires that the components
 117 in the same type are independently and identically distributed. When the components in the
 118 same subsystem are only independent, the system structure in Eq. (1) could not be directly
 119 computed, for the probability of every state vector in S_{l_1, l_2, \dots, l_K} is not the same with each other.

120 The principle of Eq. (1) is to generate all possible $\prod_{k=1}^K {m_k \choose l_k}^{-1}$ state vectors with (l_1, l_2, \dots, l_K)
 121 components of subsystem $(1, 2, \dots, K)$ functioning and then count the number of state vectors

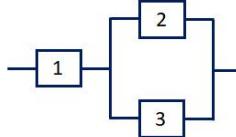
124 that result in system functioning. For the simple system with three components in Fig. 1, the
 125 survival signature $\Phi(2) = \frac{2}{3}$ when components are independently and identically distributed.
 126 Suppose the failure probability of each component is P_f , the state function and the
 127 corresponding probability are given as follows, which is also known as the exchangeability
 128 assumption,

$$P(x_1 = 1, x_2 = 1, x_3 = 0) = P(x_1 = 1, x_2 = 0, x_3 = 1) = P(x_1 = 0, x_2 = 1, x_3 = 1) = P_f \cdot (1 - P_f)^2, \quad (4)$$

$$\phi(x_1 = 1, x_2 = 1, x_3 = 0) = 1, \quad (5)$$

$$\phi(x_1 = 1, x_2 = 0, x_3 = 1) = 1, \quad (6)$$

$$\phi(x_1 = 0, x_2 = 1, x_3 = 1) = 0. \quad (7)$$



129

130 Fig.1 Simple system with 3 components

131 However, suppose components 1, 2 and 3 with different failure probability P_{f1} , P_{f2} and P_{f3} ,
 132 the probabilities of each state vector and the associated survival signature $\Phi(2)$ must be
 133 modified to the following

$$P(x_1 = X_1, x_2 = X_2, x_3 = X_3) = \prod_{i=1}^3 [(1 - P_{fi}) \cdot X_i + P_{fi} \cdot (1 - X_i)], \quad (8)$$

$$\Phi(2) = \frac{P(x_1=1, x_2=1, x_3=0) + P(x_1=1, x_2=0, x_3=1)}{P(x_1=1, x_2=1, x_3=0) + P(x_1=1, x_2=0, x_3=1) + P(x_1=0, x_2=1, x_3=1)}. \quad (9)$$

134 where $X_i (i = 1, 2, 3) = 0$ or 1 . Therefore, the survival signatures of systems with independent
 135 but nonidentical components should be calculated in the form as shown in Eq. (10), rather than
 136 Eq. (1),

$$\Phi(l_1, l_2, \dots, l_K) = \frac{\sum_{\underline{x} \in S_{l_1, l_2, \dots, l_K}} \phi(\underline{x}) \cdot P(\underline{x})}{\sum_{\underline{x} \in S_{l_1, l_2, \dots, l_K}} P(\underline{x})}. \quad (10)$$

137 where $\Phi(l_1, l_2, \dots, l_K)$ is the probability that the system functions when l_k out of m_k
 138 components function in subsystem k ($k = 1, 2, \dots, K$). To calculate the survival signature, it is
 139 necessary to obtain $\sum_{\underline{x} \in S_{l_1, l_2, \dots, l_K}} P(\underline{x})$ and $\sum_{\underline{x} \in S_{l_1, l_2, \dots, l_K}} \phi(\underline{x}) \cdot P(\underline{x})$ separately, where

140 $\sum_{\underline{x} \in S_{l_1, l_2, \dots, l_K}} P(\underline{x})$ is the probability structure. According to Eq. (1) and (10), for the same
 141 system, survival signature $\Phi(l_1, l_2, \dots, l_K)$ is no longer only related to the system function $\phi(\underline{x})$,
 142 when components are independently but nonidentically distributed, survival signature is also
 143 related to the failure probability $P(\underline{x})$ of components, which is different from the definition
 144 proposed in [11]. The system reliability can also be obtained by setting each component as a
 145 different type, then using Eq. (1). In this way there are only two conditions for each type: $l_i = 0$
 146 ($i = 1, \dots, m$), component failure and $l_i = 1$, component function. Hence, there are 2^m survival
 147 signatures must to be considered, which will be very time-consuming. This will be discussed
 148 in detail in section 6.

149 2.2 The general form of probability structure

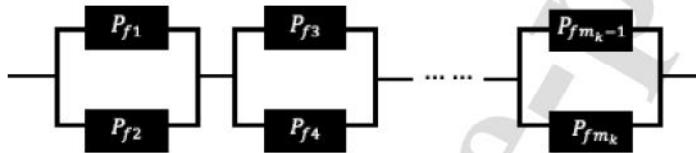
150 In Eq. (2), it's much easier to calculate probability structure when the failure probability of
 151 components is identical, as it can be transformed into the form of Bernoulli binomial
 152 distribution. However, when the failure probability of component is not equal to each other, it

153 needs to calculate $C_{m_k}^{l_k} = \frac{m_k!}{(m_k-l_k)!\cdot l_k!}$, $k \in \{1,2,\dots,K\}$ different equations, each of which
 154 represents the probability that there are $m_k - l_k$ components failure while l_k components
 155 working in the subsystem k . A simple system shown in Fig. 2 is selected as an example
 156 discussed herein. m_k components in the system k are independent. If $m_k = 4$, the calculation
 157 of $P(D_k(t) = 2)$ with *iid* assumption (each component with failure probability $P_f(t)$) and in
 158 case of INID (component i with failure probability $P_{fi}(t)$, $i = 1,2,3,4$) can be expressed as in
 159 Eq. (11) and Eq. (12), respectively.

$$P(D_k(t) = 2) = C_4^2 \cdot P_f(t)^2 \cdot (1 - P_f(t))^2, \quad (11)$$

$$\begin{aligned} P(D_k(t) = 2) &= P_{f1}(t)P_{f2}(t)(1 - P_{f3}(t))(1 - P_{f4}(t)) + P_{f1}(t)P_{f3}(t)(1 - \\ &\quad P_{f2}(t))(1 - P_{f4}(t)) + P_{f1}(t)P_{f4}(t)(1 - P_{f2}(t))(1 - P_{f3}(t)) + \\ &\quad P_{f2}(t)P_{f3}(t)(1 - P_{f1}(t))(1 - P_{f4}(t)) + P_{f2}(t)P_{f4}(t)(1 - P_{f1}(t))(1 - \\ &\quad P_{f3}(t)) + P_{f3}(t)P_{f4}(t)(1 - P_{f1}(t))(1 - P_{f2}(t)). \end{aligned} \quad (12)$$

160



161

162 Fig.2 Simple system k with m_k components

163 As can be observed, when the survival signature theory is used to evaluate a system with
 164 INID components, the formula under the *i.i.d* assumption is very different from the survival
 165 function of the system. Consequently, new techniques are needed to allow system reliability for
 166 a system with solely independent components based on the survival signature idea.

167 **3. Algorithm for computing probability structure**168 **3.1 Divide-and-conquer algorithm**

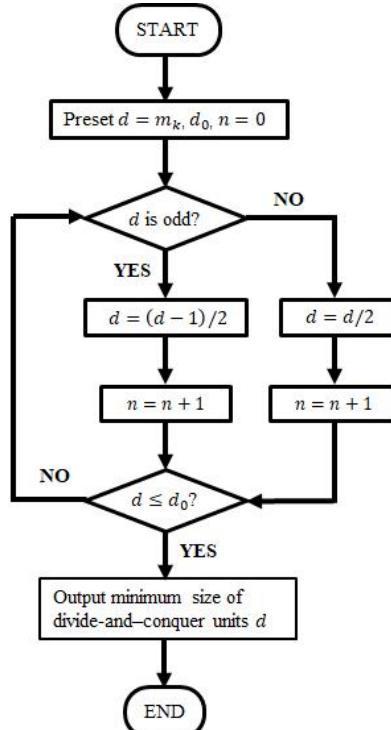
169 Divide-and-conquer [35] refers to a class of algorithm techniques in which the target
 170 problem is broken into several parts, solves the problem in each part recursively, and then
 171 combines the solutions to these subproblems into an overall solution. As shown in Section 2.2,
 172 it is quite cumbersome to solve $C_{m_k}^{l_k}$ different equations one by one when m_k or l_k gets larger.

173 Nevertheless, the computation can be much easier if it is divided into several minor groups of
 174 calculations. For example, the recursive equation based on the Bernoulli binomial in Eq. (13)
 175 shows the relationship between the system with m_k components and two divided subsystems
 176 with $\frac{m_k}{2}$ components:

$$C_{m_k}^{l_k} = \sum_{i=0}^{l_k} C_{\frac{m_k}{2}}^i \cdot C_{\frac{m_k}{2}}^{l_k-i}. \quad (13)$$

177 In consideration of the characteristics of Bernoulli polynomials and the computational
 178 efficiency, each decomposition is dichotomized. However, not every partition will divide m_k
 179 into two sub-problems of $\frac{m_k}{2}$ size. The flow chart in Fig. 3 is used to solve the size of minimum
 180 divide-and-conquer units. Here, d represents the size of the minimum divide-and-conquer units,
 181 the initial value is m_k ; d_0 represents the upper bounds of the size of the minimum divide-and-

182 conquer units, which will be discussed later in section 3.2; when the size of the minimum
 183 divide-and-conquer units is less than d_0 , no further division is performed; n represents the
 184 number of divisions.



185

186 Fig.3 Flow chart to calculate the size of minimum divide-and-conquer units

187 The above algorithm can divide m_k components into units of size d and $d + 1$. Suppose
 188 N_1 represents the number of units of size d and N_2 represents the number of units of size $d +$
 189 1, they satisfy the following relationship:

$$N_1 + N_2 = 2^n. \quad (14)$$

$$N_1 \cdot d + N_2 \cdot (d + 1) = m_k. \quad (15)$$

190 Here, an $(n + 1) \times 2^n$ (n represents the number of divisions) matrix A is utilized to store
 191 each solution in the divide-and-conquer algorithm. The probability structure can first be solved
 192 with the minimum unit of division. The $(n + 1)$ th row of the matrix A represents the
 193 probability structure of 2^n minimum divide-and-conquer units, each of which is a $1 \times (l_k + 1)$
 194 vector, and the j th element of the vector represents the probability that j components function
 195 in each minimum unit. If l_k is larger than the unit size d or $d + 1$, the probability that j ($j =$
 196 $d + 1, \dots, l_k$) components in the unit work is 0. The matrix A and $A(n+1,1)$ are shown in
 197 Fig.4a and Fig.4b.

$$\begin{array}{ccccccccc}
 & 1 & 2 & 3 & 4 & \dots & 2^{i-1} & \dots & 2^n \\
 1 & \left[\begin{matrix} \{1 \times (l_k + 1)\} \\ \{1 \times (l_k + 1)\} \{1 \times (l_k + 1)\} \\ \{1 \times (l_k + 1)\} \{1 \times (l_k + 1)\} \dots \{1 \times (l_k + 1)\} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ i & \{1 \times (l_k + 1)\} & \dots & \dots & \dots & \dots & \{1 \times (l_k + 1)\} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ n+1 & \{1 \times (l_k + 1)\} & \dots & \dots & \dots & \dots & \dots & \{1 \times (l_k + 1)\} \end{matrix} \right] & & & & & & & \\
 & 1 & 2 & \dots & l_k + 1 & & & & \\
 & P(D_{n_1}(t) = 0) & P(D_{n_1}(t) = 1) & \dots & P(D_{n_1}(t) = l_k) & & & &
 \end{array}$$

198

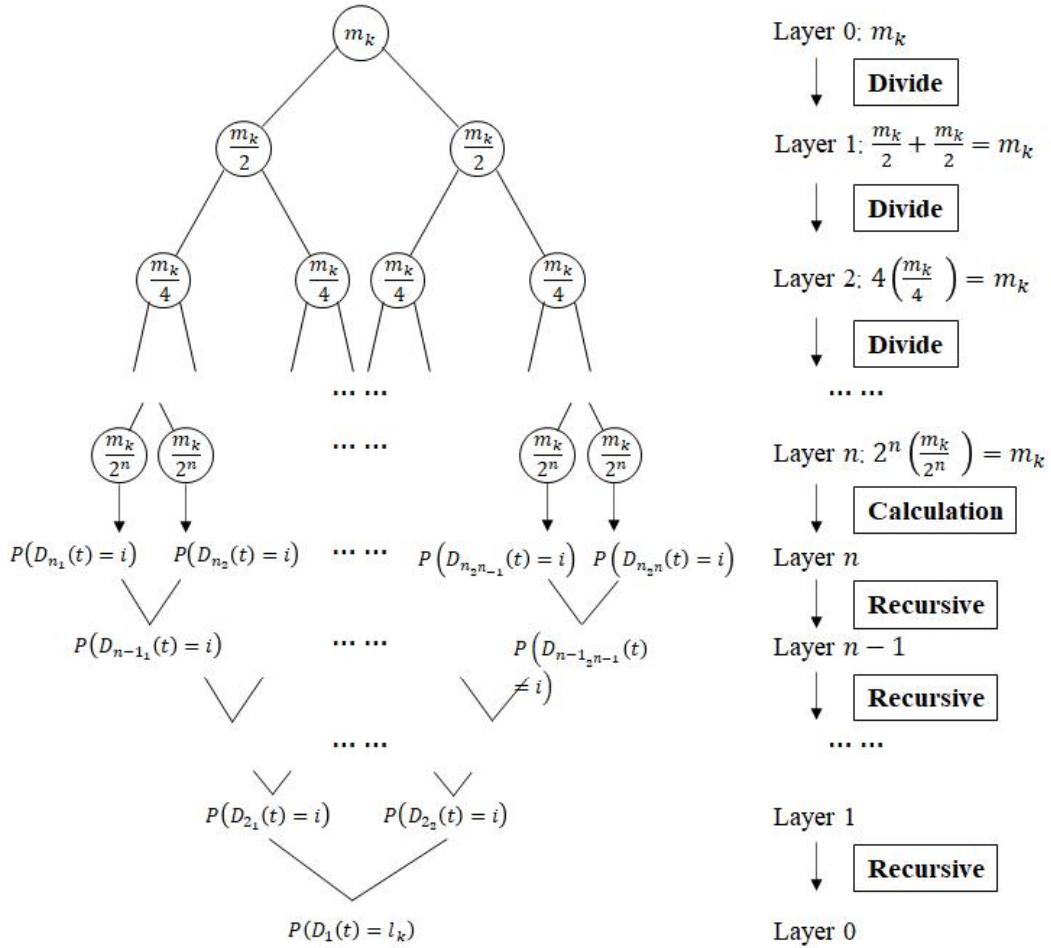
(a) Diagram of matrix A

(b) Diagram of matrix A (n + 1,1)

199 Fig.4 Diagram of the matrix A and A (n + 1,1)

200

201 Note that $P(D_{ij}(t) = m)$ ($i = 1, \dots, n$, $j = 1, 2, \dots, 2^i$, $(m = 0, 1, \dots, l_k)$) represents the
 202 probability that m components work in the j th divided unit after i divisions.



203
 204 Fig.5 Flow chart of the proposed divide-and-conquer algorithm
 205

206 The i th row of the matrix A has 2^{i-1} vectors, each of which is derived from the $(i+1)$ th
 207 row by solving Eq. (16). For example, the first vector in the n th row of the matrix A is derived
 208 from the first and second vectors in the $(n+1)$ th row, and the second vector in the n th row is
 209 derived from the third and fourth vectors in the $(n+1)$ th row, ..., the 2^{n-1} th vector in the n th
 210 row is derived from the $(2^n - 1)$ th and 2^n th vectors in the $(n+1)$ th row. The $A(i,j)$ is given
 211 as follows:

$$P(D(t) = s) = \sum_{q=0}^s P(D_{i+1_{2 \times j-1}}(t) = q) \cdot P(D_{i+1_{2 \times j}}(t) = s-q). \quad (16)$$

212 where $P(D_{ij}(t) = s)$ ($i = 1, \dots, n$, $j = 1, 2, \dots, 2^i$, $(s = 0, 1, \dots, l_k)$) is the probability that s
 213 components work in the j th divided unit after i divisions. Figure 5 shows the flow chart of the
 214 proposed method to solve probability structure. And the whole pseudocode of the proposed
 215 divide-and-conquer algorithm in computing the probability structure is shown below.
 216

Algorithm 1: Divide-and-conquer algorithm to compute probability structure

Input: m_k components in subsystem k , num of components l_k working in subsystem

k ($k = 1, \dots, K$), failure probability P_{fi} for component i ($i = 1, \dots, m$), the minimum size of divide-and-conquer units d and the number of divisions n

Output: Probability structure $P(D_k(t) = l_k)$ in subsystem k

1: $N_1 = 2^n \times (d + 1) - m_k$; $N_2 = m_k - 2^n \times d$;

2: For $i = n$: 0 do

3: If $i = n$ then

4: For $j = 1: 2^i$ do

5: If $j \leq N_1$ then

6: For $s = 0: l_k$ do

6: If $s \leq d$

7: Calculate $P(D_{i,j}(t) = s)$;

8: Elseif $s > d$

9: $P(D_{i,j}(t) = s) = 0$;

10: End-If

11: End-For

12: Elseif $j > N_1$ then

13: For $s = 0: l_k$ do

14: If $s \leq d + 1$

15: Calculate $P(D_{i,j}(t) = s)$;

16: Elseif $s > d + 1$

17: $P(D_{i,j}(t) = s) = 0$

18: End-If

19: End-For

20: End-If

21: End-For

22: Elseif $i \neq n$ then

23: For $j = 1: 2^i$ do

24: For $s = 0: l_k$ do

25: Set $P(D_{i,j}(t) = s) = 0$;

26: For $m = 0: s$ do

27: $P(D_{i,j}(t) = s) = P(D_{i+1_{2^{j-1}}}(t) = s) \cdot P(D_{i+1_{2^j}}(t) = s - m) + P(D_{i,j}(t) = s)$;

28: End-For

29: End-For

30: End-For

31: End-If

32: End-For

217

218 3.2 Illustrative example

219 A simple system shown in Fig. 2 is selected as an example study herein. The components
 220 of the subsystem k shown in Fig.2 are independent but nonidentical and the probability
 221 structure $P(D_k(t) = l_k)$ must be determined. The effect of the size of the minimum divide-and-
 222 conquer unit on the computation time of the algorithm is analyzed herein. In this paper, all the
 223 computations are performed in MATLAB on a computer with Intel Core i7-9700 of 3.00 GHz
 224 and 16 GB RAM. Suppose $m_k = 960$, $l_k = 3$, the minimum size of the divide-and-conquer
 225 unit with 480, 240, 120, 60, 30, 15, 7 and 8 components are tested separately.

226 Figure 6 shows the relationship between calculation time and the number of divisions in
 227 the divide-and-conquer algorithm. It can be seen that the minimum divide-and-conquer unit
 228 greatly influences the survival signature computation time. The calculation time of the divide-

and-conquer algorithm after 7 divisions (about 0.044273 sec) is the least. The calculation time for 7 divisions is about 2000 times shorter than for 1 division (about 108.954 sec). When the number of divisions $n = 7$, it takes approximately 0.000818 seconds for the process in Fig. 3 and approximately 0.040147 seconds to solve the probability structure for each minimum divide-and-conquer unit. This indicates that the conquering of the minimum divide-and-conquer unit, rather than the process of division and regression, takes up the majority of the running time of the divide-and-conquer algorithm in the calculation of probability structure. And when we perform 8 divisions and the size of the minimum divide-and-conquer unit is 3 and 4, the running time of the algorithm is 0.026962 sec, and the efficiency of the overall improvement is not high. Therefore, the upper limit of the minimum partition and conquer unit $d_0 = 10$ could obtain better operation efficiency.

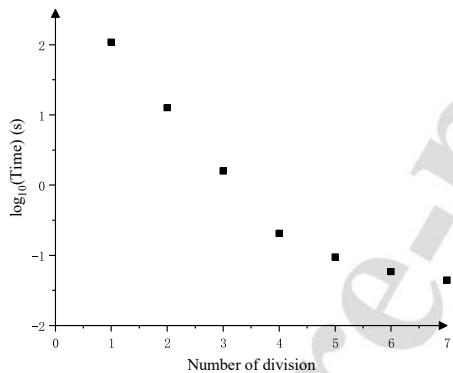
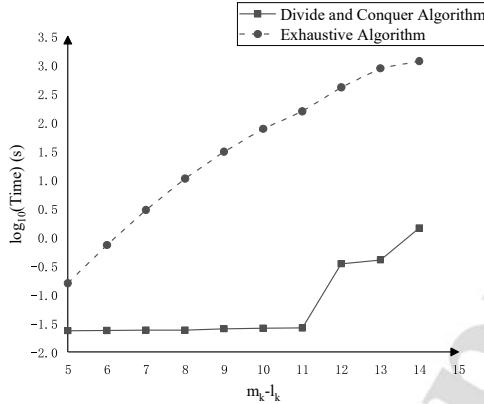


Fig.6 Relationship between calculation time and the number of divisions in the divide-and-conquer algorithm

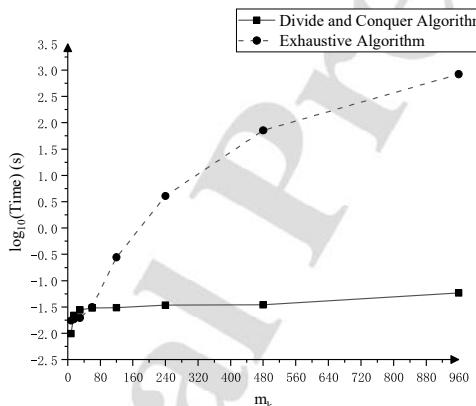
To prove the effectiveness of the algorithm, the divide-and-conquer algorithm is compared to the exhaustive algorithm. As mentioned in Section 2.2, there are $C_{m_k}^{l_k} = \frac{m_k!}{(m_k - l_k)! \cdot l_k!}$ cases where l_k components working in a m_k components system with different probabilities. The exhaustive algorithm refers to iterating over all the cases and calculating the probability of each case, then adding them up. By increasing m_k or l_k , the exhaustive algorithm leads to dramatic computations, for it will take a long time to iterate all $C_{m_k}^{l_k}$ cases. The comparison of calculation time between the exhaustive algorithm and the proposed algorithm for different numbers of failure components $m_k - l_k$ when $m_k = 30$ with the number of divisions equal to 2 is shown in Fig. 7. For the number of failure components $m_k - l_k = 3$, the comparison of calculation time between the exhaustive algorithm and the proposed algorithm for different m_k is shown in Fig. 8.

In Fig. 7, when $m_k = 30$ and $l_k = 14$, the average computation time of the exhaustive algorithm is 1161.048s, and the average time of the divide-and-conquer algorithm is 1.4413s. In Fig. 8, when $m_k = 960$ and the number of failed components is 3 ($m_k - l_k = 3$), the average time of the exhaustive algorithm is 835.43s, the average time of the divide-and-conquer algorithm is 0.05826s. As can be seen from Fig. 7 and Fig. 8, the divide-and-conquer algorithm is highly efficient in estimating the probabilistic structures for large systems. For example, for

261 a system with $m_k = 10000$ and $l_k = 5000$, the exhaustive procedure cannot be executed in
 262 the meager 170s of processing time required by the divide-and-conquer algorithm. In terms of
 263 data usage, the divide-and-conquer algorithm is only related to the number of divisions n and
 264 the number of failure components l_k . And 2^{n-1} recursive computations from $(n+1)$ th row to
 265 the n th row of the matrix A , 2^{n-2} recursive computations from the n th row to the $(n-1)$ th
 266 row..., so the whole process needs $(2^n - 1) \times l_k$ recursive calculation.



267
 268 Fig.7 Comparison between the exhaustive algorithm and divide-and-conquer algorithm in the
 269 calculation time for $m_k = 30$ when the number of divisions $n = 2$



270
 271 Fig.8 Comparison between exhaustive algorithm and divide-and-conquer algorithm in
 272 calculation time when the number of failure components is 3
 273

274 4. Algorithm for computing system structure

275 Based on the divide-and-conquer algorithm, a Monte Carlo simulation-based approach is
 276 further developed to calculate the survival signatures. According to Eq. (10), when there are l_k
 277 components working in subsystem k , the probability of each state vector in the approach might
 278 be different from each other under the INID situation. Therefore, it can be sampled from the
 279 probability density of the different state vectors and count the number of vectors that lead to
 280 the system function to obtain the survival signatures. The approach is carried out in two steps:
 281 Sampling N dataset from the set S_{l_1, l_2, \dots, l_K} of all state vectors for the whole system when l_k
 282 components working in the subsystem k ($k = 1, 2, \dots, K$) based on the probability of each state
 283 vector; determining the system state function $\phi(\underline{x})$ for each generated sample. This is
 284 elucidated in more detail below.

285 4.1 The crude Monte Carlo simulation

286 To determine the system structure, the crude Monte Carlo sampling method is the most
 287 direct way which is used as shown in **Algorithm 2**. The reason why this sampling method is
 288 crude is that it needs to sample all the state vector sets (N_{all} in **Algorithm 2**) according to the
 289 failure probability of each component, and then select the state vectors (N in **Algorithm 2**)
 290 satisfying the work of l_k components in the subsystem k . The number of samples N obtained
 291 by this sampling method is not stable. If N_{all} is too large, the number of samples N at time t is
 292 approximately equal to $N_{all} \times \prod_{k=1}^K P(D_k(t) = l_k)$. If a large N is used to calculate $\phi(\underline{x})$, a
 293 larger N_{all} is required, which is very time-consuming and inefficient when the failure
 294 probability of components is very small, that is, $\prod_{k=1}^K P(D_k(t) = l_k)$ is very small. Therefore,
 295 we need a more efficient sampling method to avoid producing invalid samples
 296 ($\underline{x} = (x_1, x_2, \dots, x_m) \notin S_{l_1, l_2, \dots, l_K}$).
 297

Algorithm 2: Crude Monte Carlo sampling method

Input: The failure probability P_{fi} ($i = 1, 2, \dots, m$) at time t for each component, the working
 components number l_k for each subsystem k ($k = 1, 2, \dots, K$), N_{all}

Output: N samples from the set S_{l_1, l_2, \dots, l_K}

Preset: $N = 0$

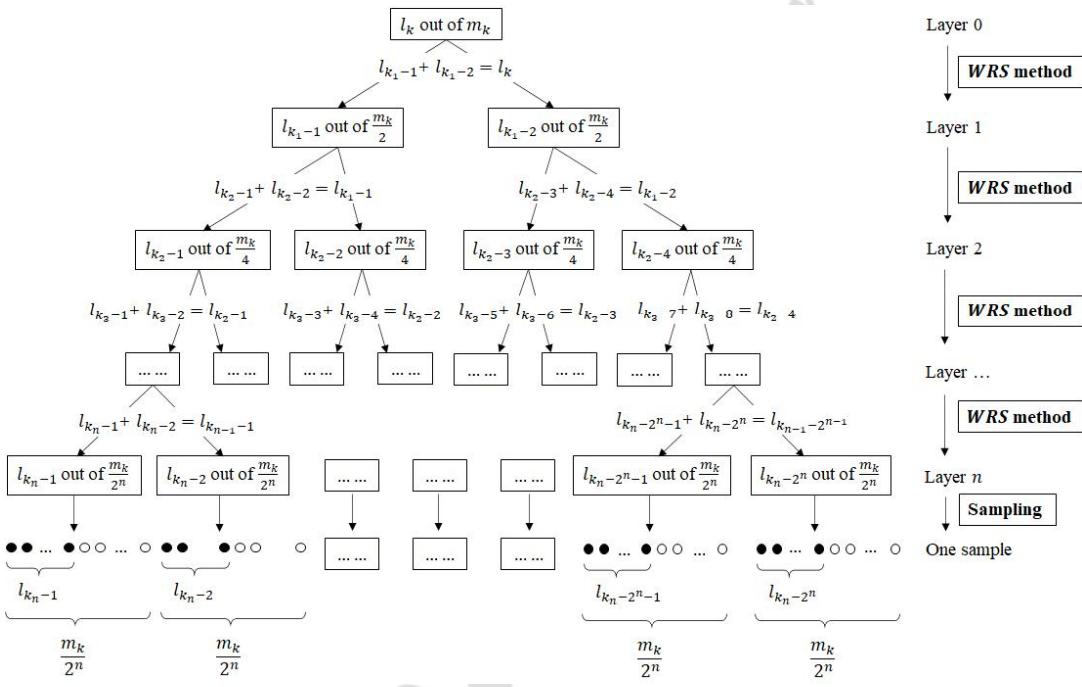
- 1: For $n = 1 : N_{all}$ do
- 2: For $i = 1 : m$ do
- 3: $u_i = \text{random}(0,1);$
- 4: If $u_i \leq P_{fi}$
- 5: $X_i = 0;$
- 6: Elseif $u_i > P_{fi}$
- 7: $X_i = 1;$
- 8: End-If
- 9: End-For
- 10: If $\underline{x} = (x_1, x_2, \dots, x_m) \in S_{l_1, l_2, \dots, l_K}$
- 11: $N = N + 1;$
- 12: End-If
- 13: End-For

298

299 4.2 New simulation method for survival signature

300 As mentioned in the previous section, when m_k is large, it is difficult to obtain a sample
 301 that has exactly l_k working properly. However, if m_k is divided into 2^n parts and L_i ($i =$
 302 $1, 2, \dots, 2^n$) components work in each part, the sampling efficiency will be much higher. In this
 303 way, a sample that has exactly l_k working components can be obtained when $\sum_{i=1}^{2^n} L_i = l_k$. The
 304 key to the whole process is how to randomly determine the number of working components in
 305 each part so that the number of all normal components is l_k . As mentioned in Section 3.1,
 306 computing the probability of l_k components working in subsystem k can be converted into the
 307 combination of computing two probabilistic structures containing $\frac{m_k}{2}$ components assuming
 308 m_k is even, as shown in Eq. (16). For a sample with l_k components working in the subsystem
 309 k , if we divide the m_k once, there will be $l_k + 1$ cases, that is, q ($q = 0, 1, \dots, l_k$) components
 310 working in the first divided part containing $\frac{m_k}{2}$ components while $l_k - q$ components working
 311 in the second divided part containing $\frac{m_k}{2}$ components. If we continue to divide the first divided

312 part containing $\frac{m_k}{2}$ components with q ($q = 0, 1, \dots, l_k$) components working into 2 subparts,
 313 there will be another $q + 1$ cases where r ($r = 0, 1, \dots, q$) components working in the first
 314 divided subpart containing $\frac{m_k}{4}$ components and $q - r$ components working in the second
 315 divided subpart containing $\frac{m_k}{4}$ components. Therefore, in each Monte Carlo sampling, the main
 316 step is to determine how many components working in each of the two subparts formed by each
 317 division, as shown in Fig. 9. A matrix B (Fig.10) is used to store the number of working
 318 components in each partition unit of a sample, where l_{k_i-j} ($i = 1, \dots, n$, $j = 1, 2, \dots, 2^i$)
 319 represents the number of components working in the j th subsystem after i divisions for one
 320 generated sample, $l_{k_i-j} + l_{k_i-j+1} = l_{k_{i-1}-\frac{j+1}{2}}$ ($j = 1, 3, 5, \dots, 2^i - 1$).



321
 322 Fig.9 Flow chart of one generated l_k -out-of- m_k sample
 323 The matrix B is generated from top to bottom, $B(i+1, 2j)$ ($i = 1, 2, \dots, n; j = 1, 2, \dots, 2^{i-1}$) and $B(i+1, 2j+1)$ are randomly sampled from $B(i, j) + 1$ cases in which
 324 $B(i, j)$ components working in the j th unit after $i - 1$ divisions. However, the weight of each
 325 case is different, which is related to the number $B(i, j)$ of working components. The calculation
 326 formula is given as

328

$$\begin{array}{c}
 & 1 & 2 & 3 & 4 & \dots & 2^{i-1} & \dots & 2^n \\
 \begin{matrix} 1 \\ 2 \\ 3 \\ \vdots \\ i \\ \vdots \\ n+1 \end{matrix} & \left[\begin{array}{ccccccc}
 l_k & & & & & & \\
 l_{k_1-1} & l_{k_1-2} & & & & & \\
 l_{k_2-1} & l_{k_2-2} & l_{k_2-3} & l_{k_2-4} & & & \\
 \vdots & \vdots & \vdots & \vdots & \ddots & & \\
 l_{k_{i-1}-1} & \cdots & \cdots & \cdots & \cdots & l_{k_{i-1}-2^{i-1}} & \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\
 l_{k_n-1} & \cdots & \cdots & \cdots & \cdots & \cdots & l_{k_n-2^n}
 \end{array} \right]
 \end{array}$$

329

 Fig.10 Diagram of matrix B of one generated sample

$$\omega_{q+1} = \frac{P(D_{i_2 \times j-1}(t)=q) \cdot P(D_{i_2 \times j}(t)=B(i,j)-q)}{\sum_{z=0}^{B(i,j)} P(D_{i_2 \times j-1}(t)=z) \cdot P(D_{i_2 \times j}(t)=B(i,j)-z)}, \quad (17)$$

330 where ω_{q+1} represents the $(q + 1)$ th case weight ($q = 0, 1, \dots, B(i, j)$), $B(i, j)$ represents the
 331 number of working components in the j th unit after $i - 1$ divisions. So, the generation of the
 332 matrix B is a process of weighted random sampling (WRS) which is to randomly select m
 333 elements in a set V with n elements, $v_i \in V$ with a weight ω_i , and the probability of each
 334 element being selected is the ratio of the weight. Efraimidis [36],[37] proposed the weighted
 335 random sampling algorithm to solve this problem, herein, the process of solving for
 336 $B(i + 1, 2j)$ ($i = 1, 2, \dots, n; j = 1, 2, \dots, 2^{i-1}$) and $B(i + 1, 2j + 1)$ transforms into select a
 337 case from a set V with $B(i, j) + 1$ cases, each case having a weight ω_{q+1} ($q = 0, 1, \dots, B(i, j)$).
 338 Weighted sampling has also been widely used in reliability analysis [38].

339 Each element in the probability structure matrix A (Fig.3a) calculated by the divide-and-
 340 conquer algorithm represents the probability that i ($i = 0, 1, \dots, l_k$) components work in the
 341 divided part. Each item in Eq. (17) can be obtained from the matrix A to avoid repeated
 342 calculations. Different from the bottom-up calculation process of the divide-and-conquer
 343 algorithm, this sampling method is top-down. By selecting the generation mode of each layer,
 344 the last generated samples of the bottom layer are combined into the final samples. Therefore,
 345 there will be no invalid samples. The corresponding pseudocode is shown in **Algorithm 3**.
 346

Algorithm 3: Sampling N state vectors for l_k -out-of- m_k system

Input: Number of system components m_k ; the number of working components l_k ; the number
 of targets generated samples N ; divide and conquer algorithm calculated matrix A

Output: N state vectors for l_k -out-of- m_k system

Preset: the size of matrix B = the size of matrix A , $B(1,1) = l_k$

1: For $i = 1$ to N do

2: For $j = 1$ to n do (n is the number of divisions, obtained from size A)

3: For $k = 1$ to 2^{j-1} do

4: $\omega_{q+1} = \frac{P(A\{j+1,2 \times k\}(1,q+1)) \cdot P(A\{j+1,2 \times k+1\}(1,B(j,k)-q+1))}{\sum_{q=0}^{B(j,k)} P(A\{j+1,2 \times k\}(1,q+1)) \cdot P(A\{j+1,2 \times k+1\}(1,B(j,k)-q+1))}$ ($q = 0, 1, \dots, B(j, k)$);

5: $B(j+1,2 \times k) = WRS$ for set $\{0, 1, \dots, B(j, k)\}$ with weight $\omega = \{\omega_1, \dots, \omega_{q+1}\}$;

6: $B(j+1,2 \times k + 1) = B(j, k) - B(j+1,2 \times k)$;

7: End-for

8: End-For

9: For $q = 1$ to 2^n do

10: WRS for q th minimum divide-and-conquer unit with $B(n, q)$ components working;

11: End-For

12: End-For

347

To obtain $\phi(\underline{x})$, one must determine the failure modes for the system. The link between the system's components is converted into a directed graph in order to determine the state vector in a traditional manner. The system structure can be replaced by a graph structure if the structure's constituent parts are transformed into nodes and the relationships among them into edges. The root node S and terminal node T can be used to describe the system status. A path is defined as a set of system components so that if these components are failure-free, the system is up. Therefore, for a certain state vector \underline{x} , the system state function $\phi(\underline{x})$ can be determined by finding whether there is a path from the root node to the terminal node. This algorithm has been used in the calculation of survival signature, and the *computeSystemSurvivalSignature()* function in the *ReliabilityTheory* R package is based on this graph model [29].

The binary decision diagram (*BDD*) method is also one way to determine $\phi(\underline{x})$. In a *BDD* structure, the 0-edge or 1-edge is used to connect nodes, where the 0-edge indicates the failure of the previous node and the 1-edge indicates the function of the previous node. There are two terminal nodes 0-node and 1-node, which means the system fails when the state vector \underline{X} leads to 0-node and system functions when the state vector \underline{X} leads to 1-node. Feng et.al [31] and Reed [32],[33] have proposed efficient algorithms to implement binary decision diagrams, multidimensional arrays and dynamic programming paradigms in survival signature. However, it is tedious to calculate the minimum cut set of a directed graph or construct a *BDD* structure for a system [39]. Therefore, in the actual system analysis, it is necessary to flexibly choose different methods to obtain the failure modes according to the different systems.

Algorithm 4: Compute $\Phi(l_1, \dots, l_K)$

Input: generated N samples with l_k components working in subsystem k ($k = 1, 2, \dots, K$), system failure modes (the *BDD* structure, directed graph for system, or other methods)

Output: $\Phi(l_1, \dots, l_K)$

1: Set $N_{work} = 0$;

2: For $n = 1$ to N do

3: Obtain the system state function $\phi(\underline{x}_n)$ the generated $\underline{x}_n = (x_1, \dots, x_m)$;

4: If $\phi(\underline{X}_n) = 1$, $N_{work} = N_{work} + 1$, End-If

5: End-For

6: $\Phi(l_1, \dots, l_K) = \frac{N_{work}}{N}$.

368

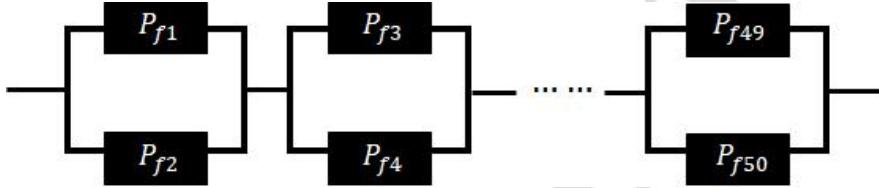
By integrating **Algorithm 1**, **Algorithm 3** and **Algorithm 4**, there are four steps to obtain survival signature: firstly, calculate the size and number of divided units; secondly, calculate the probability structure of each divided unit and recursive for the system probability structure; thirdly, top-down weighted random sampling based on the probability structure; finally, obtain the system state function according to system topology structure and calculate the survival signature.

374 4.3 Illustrative example

375 To illustrate the developed algorithm, a series-parallel system with m_k components is
 376 analyzed as an example study, as shown in Fig. 11 ($m_k = 50$). The new simulation method and
 377 the crude Monte Carlo simulation method are used to generate a total of 10,000 samples. Six
 378 groups of state vectors are tested, these are 40-out-of-50 ($m_k = 50$, $l_k = 40$), 25-out-of-50,
 379 90-out-of-100, 50-out-of-100, 190-out-of-200 and 100-out-of-200 samples.

380 Table 1 shows the sampling completion time of the two sampling methods. As can be seen
 381 from Table 1, the new simulation method greatly improves the calculation time compared to

the crude Monte Carlo method. And with the increase in the total number of components m_k and the number of failed components, the crude Monte Carlo method will not be able to collect samples effectively. This is mainly because the crude Monte Carlo needs to judge whether each generated sample belongs to the set S_{l_1, l_2, \dots, l_K} , which will inevitably produce invalid samples that do not meet the conditions, leading to the reduction of sampling efficiency. If the reliability of components decreases or the m_k and l_k of the system increases, the number of effective samples decreases and the number of effective samples from Monte Carlo simulations is about $N_{all} \times P(D_k(t) = l_k)$. When N_{all} remains unchanged and $P(D_k(t) = l_k)$ decreases, the number of effective samples decreases. It takes about 12 seconds for the crude Monte Carlo method to generate 10,000 valid samples for a total of 180,000 samples when $m_k = 50$, $l_k = 40$. For the new simulation method, the main work of the algorithm is to construct the matrix B of each sample, that is, to solve the weight of each case in each step of the weighted random sampling. Therefore, the running time of the algorithm is roughly $O(2^n \cdot l_k)$, where n denotes the number of divisions and l_k represents the number of working components in the subsystem k .



397 Fig.11 Series-parallel system with 50 components

398 Table.1 – Comparison between the new simulation method and crude Monte Carlo

Total number of samples	Execution time [sec]	
	Crude Monte Carlo	New simulation method
$m_k = 50, l_k = 25$	-	3.057
$m_k = 50, l_k = 40$	11.456	2.196
$m_k = 100, l_k = 50$	-	7.278
$m_k = 100, l_k = 90$	30.240	3.590
$m_k = 200, l_k = 100$	-	14.185
$m_k = 200, l_k = 190$	-	7.705

400 Next, the newly developed approach is tested for analyzing components having time-varying reliability properties. It is assumed that the failure probability of each component
401 follows the Gamma distribution over time as given in Eq. (18) [13]. Intervals are used to
402 describe the imprecision in the failure time distribution, with distribution parameter α in
403 [1.2,1.8] and distribution parameter β in [2.3,2.9].

$$404 p_f(t) = \frac{\beta^\alpha}{\Gamma(\alpha)} t^{\alpha-1} e^{-\beta t}. \quad (18)$$

405 where α and β are the shape parameter and size parameter for Gamma distribution, $\Gamma(\alpha)$ is the
406 Gamma function while $\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} e^{-x} dx$. The state vector generated by the new
407 simulation method is converted into the *BDD* structure, and the values of survival signatures
408 for different states are calculated and shown in Fig. 16. For a series-parallel system with m_k
409 components, it is easy to estimate the survival signature of l_k components working under the
410 i.i.d assumption as shown in Eq. (19).

$$\Phi(l_k) = \frac{m_k \cdot (m_k - 1) \cdot (m_k - 2) \cdots (m_k - 2(l_k - 1))}{m_k \cdot (m_k - 1) \cdots (m_k - l_k + 1)}. \quad (19)$$

411 Instead of the *BDD* structure, the system functions of each state vector can be obtained

more simply by determining whether each parallel component fails simultaneously. Figure 12 shows the variation of survival signatures ($\Phi(45)$, $\Phi(40)$, $\Phi(35)$) over time when each component is independent but not identically distributed and the corresponding values of survival signatures under the i.i.d assumption (the solid lines shown in Fig. 12, in the i.i.d condition for a series-parallel system with 50 components, $\Phi(45) = 0.8024$, $\Phi(40) = 0.3258$, $\Phi(35) = 0.04758$). It can be found that the survival signatures are no longer a fixed value as components are not identically distributed, which leads to different failure modes.

It should be realized that the probability structure obtained by the divide-and-conquer algorithm (**Algorithm 1**) is the analytical solution, and the system structure obtained by **Algorithm 4** is the simulated solution. To exclude the possibility that the difference in the survival signature under the two conditions in Fig. 12 is due to the randomness of Monte Carlo sampling, five reliability calculation methods of the series-parallel system are compared.

1. Analytical method $r_1(t)$: consider each parallel part as a subsystem, and the failure probability of each subsystem is calculated based on Eq. (20). The system reliability is the product of the working probability of each subsystem based on Eq. (21);
2. Survival signature method $r_2(t)$: system structure is calculated under the assumption of i.i.d with Eq. (1), and the probability structure is calculated by **Algorithm 1**;
3. Survival signature method $r_3(t)$: system structure calculated by **Algorithm 4** by generating 10,000 samples using Eq. (11), and the probability structure calculated by **Algorithm 1**.
4. Survival signature method $r_4(t)$: system structure calculated by **Algorithm 4** by generating 1,000 samples using Eq. (11), and the probability structure calculated by **Algorithm 1**.
5. Survival signature method $r_5(t)$: system structure calculated by **Algorithm 4** by generating 5,000 samples using Eq. (11), and the probability structure calculated by **Algorithm 1**.

The calculation results utilizing the above methods are shown in Fig. 13. Table 2 shows the system reliability in each year calculated by each method. Table 2 also summarizes the calculation errors according to the following formulas:

$$p_{f\text{subsystem } i}(t) = (1 - p_{f2i-1}(t)) \cdot (1 - p_{f2i}(t)), \quad (20)$$

$$r_1(t) = \prod_{i=1}^{25} (1 - p_{f\text{subsystem } i}(t)), \quad (21)$$

where $P_{fj}(t)$ ($j = 1, 2, \dots, 50$) represents the failure probability of component j , $p_{f\text{subsystem } i}(t)$ ($i = 1, 2, \dots, 25$) represents the time-dependent failure probability for each subsystem i with two parallel components.

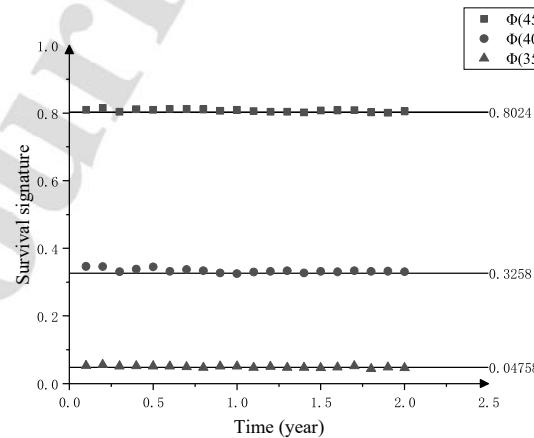
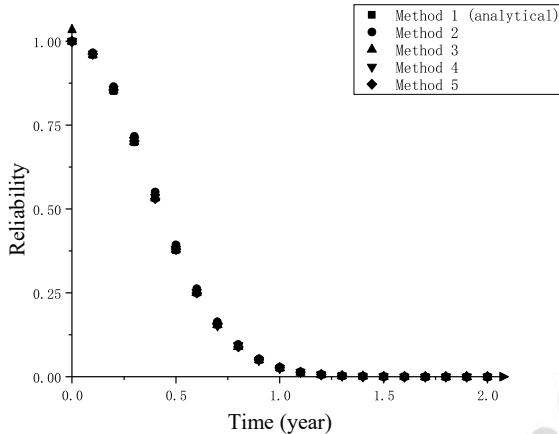


Fig.12 Survival signatures calculated by new simulation method

443



444

445

Fig.13 Comparison of reliability calculated by three methods

446 Table.2 System failure probability calculated by each method and calculation error

Time (year)	$r_1(t)$ (Analytical)	$r_2(t)$	$r_3(t)$	$r_4(t)$	$r_5(t)$
0	1	1(0%)	1(0%)	1(0%)	1(0%)
0.1	0.95917	0.96540(0.6495%)	0.95969(0.0542%)	0.96285(0.3837%)	0.96172(0.2659%)
0.2	0.85132	0.86477(1.5799%)	0.85341(0.2455%)	0.85822(0.8105%)	0.85522(0.4581%)
0.3	0.69936	0.71691(2.5094%)	0.70063(0.1816%)	0.70619(0.9766%)	0.70313(0.5391%)
0.4	0.53286	0.55081(3.3686%)	0.53321(0.0657%)	0.53558(0.5105%)	0.53112(0.3265%)
0.5	0.37777	0.39343(4.1453%)	0.37997(0.5824%)	0.38048(0.7174%)	0.38013(0.6247%)
0.6	0.25018	0.26226(4.8285%)	0.25050(0.1280%)	0.25209(0.7635%)	0.24976(0.1679%)
0.7	0.15540	0.16384(5.4311%)	0.15535(0.0322%)	0.15298(1.5573%)	0.15778(1.5315%)
0.8	0.09091	0.09633(5.9619%)	0.08966(1.3750%)	0.09323(2.5520%)	0.08957(1.4740%)
0.9	0.05029	0.05352(6.4227%)	0.05022(0.1392%)	0.04946(1.6504%)	0.05065(0.7158%)
1.0	0.02641	0.02821(6.8156%)	0.02623(0.6816%)	0.02570(2.6884%)	0.02697(2.1204%)
1.1	0.01321	0.01416(7.1915%)	0.01301(1.5140%)	0.01423(7.7214%)	0.01340(1.4383%)
1.2	0.00632	0.00312(7.4367%)	0.00623(1.4241%)	0.00661(4.5886%)	0.00645(2.0570%)

447 As can be seen from Table 2, the probability structure calculated by **Algorithm 1** is an
 448 analytical solution, therefore, the results indicate that the survival signatures are no longer a
 449 fixed value in the case of INID components. Also, it can be shown that the reliability produced
 450 by the new simulation approach (method 3, with 10,000 samples) is the one that comes closest
 451 to the analytical solution, indicating that by increasing the number of samples, the influence of
 452 sampling fluctuation may be lessened.

5. Importance index

453 The results of the calculated system reliability could provide evaluations of the component's
 454 importance for preventive maintenance or periodic inspection. The general importance index I_i
 455 of components for the binary system can be calculated as follows [40]
 456

$$I_i = P(\phi(X) = 1|X_i = 1) - P(\phi(X) = 1|X_i = 0), \quad (22)$$

457 where I_i represents the importance index for component i , $P(\phi(\underline{X}) = 1|X_i = 1)$ represents
 458 how likely the system is functioning while component i is working, $P(\phi(\underline{X}) = 1|X_i = 0)$
 459 represents how likely the system is functioning while component i is not working. Similarly,
 460 the total system reliability $P(\phi(\underline{X}) = 1)$ can be expressed as

$$P(\phi(\underline{X}) = 1) = (1 - P_{fi}) \cdot P(\phi(\underline{X}) = 1|X_i = 1) + P_{fi} \cdot P(\phi(\underline{X}) = 1|X_i = 0). \quad (23)$$

461 where P_{fi} represents the failure probability of component i . The developed algorithm can be
 462 implemented into the calculation of the importance index of a specific component i in a system
 463 with m components and K subsystems. The input of **Algorithm 1** and **Algorithm 4** is the failure
 464 probability of each component. Therefore, when calculating the system reliability
 465 $P(\phi(\underline{X}) = 1|X_i = 1)$ ($P(\phi(\underline{X}) = 1|X_i = 0)$) of component i working (failure) separately, the
 466 failure probability of corresponding component should be entered as 1 (0). In this way, in the
 467 probability structure of the minimum divide-and-conquer unit, only the probability of cases
 468 when component i is working is retained, and the probability of failure of component i is 0.
 469 Similarly, in the simulation method, to sample N state vectors for l_k -out-of- m_k system with
 470 component i being working, the sample of component failure will not be extracted because the
 471 weight corresponding to the failure of component i is 0 in each step of weighted random
 472 sampling. The smallest divide-and-conquer unit is unchanged when calculating these
 473 probabilities. It is noticed that the calculation of the importance index only considers the
 474 changes in component i , that is either work or fail, and there is no need to rebuild the new *BDD*
 475 structure or change the minimum unit of the divide-and-conquer algorithm. The calculation
 476 process is greatly simplified in an efficient way. The flow chart for calculating the importance
 477 index of component i is shown in Fig.14.

478

479 6. Example Study —— Tsinghua Campus Pipeline Network

480 To demonstrate the developed approaches in analyzing system reliability, this section
 481 considers a case study on a selected pipeline system from Tsinghua University. The
 482 underground water supply pipeline network in the Tsinghua university campus covers about
 483 300,000 m^2 . The Building Information Modeling (BIM) model for this pipeline system is
 484 shown in Fig.15. In this calculation, the total number of pipeline components is 1366 and the
 485 inner diameter of the pipeline has seven different types: 200mm, 150mm, 100mm, 80mm,
 486 65mm, 50mm and 32mm. The maximum length of the pipeline is 191.57m and the minimum
 487 length is 0.114m. The specific information of each pipe can be extracted from the REVIT model,
 488 including the inner diameter, wall thickness and toughness. By modeling water flows in the
 489 pipes using EPANET's pipeline model, it is possible to determine the working hydraulic
 490 pressure of each pipe [41]. Figure 16 depicts the hydraulic analysis model for the Tsinghua
 491 pipeline network system.

492 6.1 Component reliability analysis

493 In this study, the failure mechanism of pipe toughness under corrosion is investigated in the
 494 reliability analysis [42]. There are four failure modes considered in the pipeline, namely
 495 external hoop corrosion, internal hoop corrosion, external axial corrosion and internal axial
 496 corrosion [43], see Fig.17. Laham [44] gives the following formulas for calculating axial and
 497 hoop stress intensity factors:

$$K_{I-h} = \sqrt{\pi a} \sum_{i=0}^3 \sigma_i f_i \left(\frac{a}{d}, \frac{2c}{a}, \frac{R}{d} \right), \quad (24)$$

$$K_{I-a} = \sqrt{\pi a} \left(\sum_{i=0}^3 \sigma_i f_i \left(\frac{a}{d}, \frac{2c}{a}, \frac{R}{d} \right) + \sigma_{bg} f_{bg} \left(\frac{a}{d}, \frac{2c}{a}, \frac{R}{d} \right) \right), \quad (25)$$

where K_{I-h} denotes the stress intensity factor under hoop stress, K_{I-a} denotes the stress intensity factor under axial stress. a denotes the crack depth, d denotes the wall thickness, $2c$ denotes the crack length, R denotes the inner diameter. Geometry functions f_i and f_{bg} depends on the value of $\frac{a}{d}$, $\frac{2c}{a}$ and $\frac{R}{d}$, and could be found in different tables presented by Laham [44].

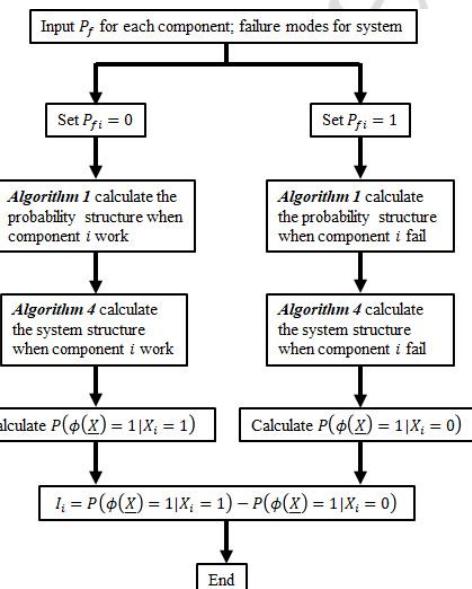
Then the time-dependent limit state function is defined in the following equation

$$G(K_{IC}, K_I, t) = K_{IC} - K_I(t), \quad (26)$$

where $G(K_{IC}, K_I, t)$ denotes the toughness limit-state function at time t , K_{IC} is the fracture toughness, the property information of the pipeline, $K_I(t)$ represents the stress intensity factor under the hoop or axial stress calculated at time t . The deterministic corrosion model is used to simulate the influence of corrosion on the pipeline reliability. A widely used model of corrosion rate between the depth of corrosion pit a and pipe age t proposed by Kucera and Mattsson [45] is given as follows:

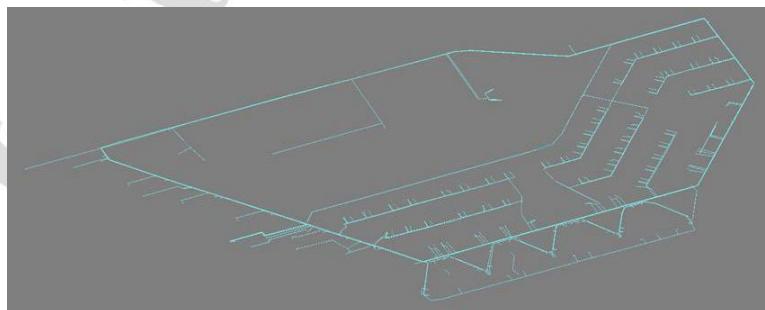
$$a = kt^n. \quad (27)$$

The empirical coefficients (k and n) are obtained by fitting the model to the observation data. The Monte Carlo simulation is adopted to calculate component reliability, and the variables and empirical coefficients calculated in Eq. (25) and Eq. (27) can be extracted from the BIM model or in the form of random variables, as shown in Appendix A.



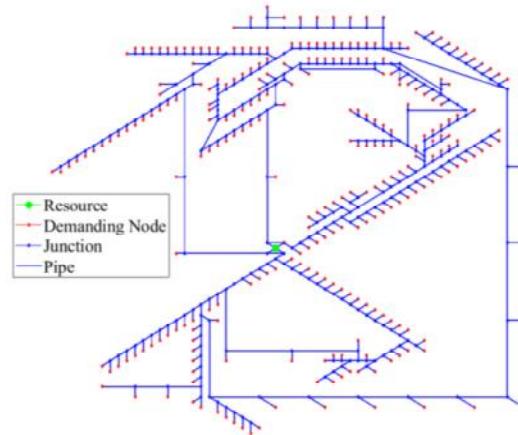
513
514

Fig.14 Flow chart of calculating importance index for component i



515
516

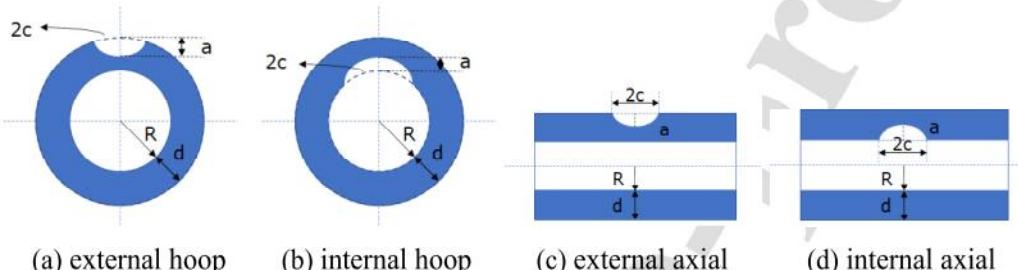
Fig.15 Tsinghua university pipeline network BIM model



517
518

Fig.16 Hydraulic analysis model of the Tsinghua pipeline network

519
520



521

Fig.17 Four failure modes for the corrosion model

522 Based on the developed algorithms, the time-dependent reliability of each pipe component
523 under the influence of corrosion is calculated and plotted in Fig.18. Here, the pipeline system
524 is considered for service for 30 years. Pipes with a failure probability greater than 0.3 are shown
525 in red lines.

526 It can be seen that a total number of 333 pipes are greatly corroded. The mean failure
527 probability of all pipes is 0.3221 with a variation of 0.0205. For all pipeline components, the
528 maximum failure probability is 0.6332 and the minimum failure probability is 0.0440 after 30
529 years of service. It would be seen that the failure probability of a larger group of pipes is greater
530 than 0.3 taking a percentage of 25%. Even nearly 16% of pipes have a very large failure
531 probability greater than 0.5.

532

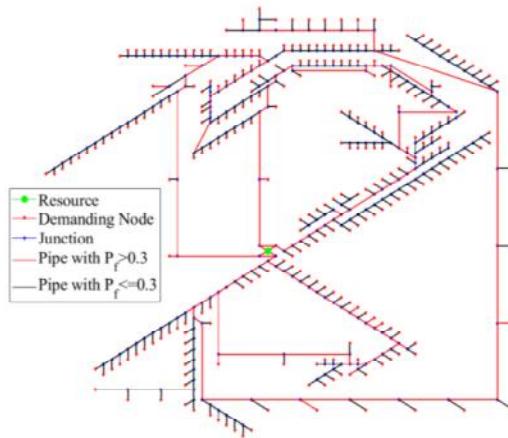
For the pipeline system that is in service for 50 years, the results are plotted in Fig.19.

533 Compared to the case of 30 years, the pipeline system of 50 years is much worse. A larger
534 percentage of corroded pipes appeared in the figure at 78.13%. A total number of 1082
535 components probabilities are greater than 0.3. The mean failure probability of all pipes is 0.3868
536 with a variation of 0.0211. And the maximum failure probability is 0.6939 and the minimum
537 failure probability is 0.0659 after 50 years of service. By the 50th year, 22% of components
538 have a failure probability of more than 0.5, which indicates that the safety degradation of
539 pipeline components is very serious.

540
541
542
543
544
545

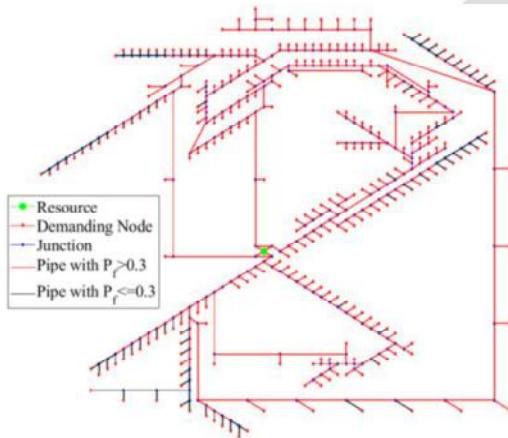
Also, it has been shown that the pipelines with a high failure chance are mainly located close to the water source. The majority of the pipelines in this area have big diameters, high operating water pressure, and extensive lengths. The pipeline with the highest and lowest likelihood of failure for the 50-year-old pipeline system is depicted in Fig. 20. As can be seen from Fig.20, the failure probability of the pipeline increases rapidly in the initial stage of service, but slows down after the 10th year. This is mainly because the exponential corrosion model is

546 used.



547

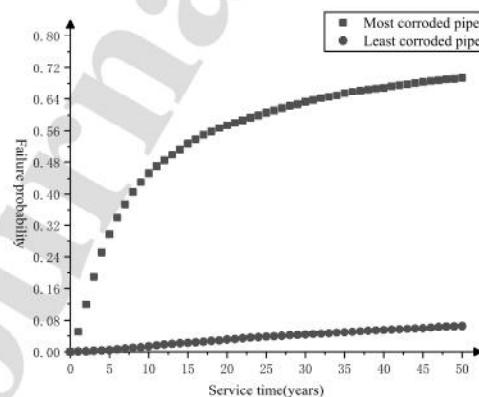
548 Fig.18 Working status of pipe network components after 30 years of service



549

550 Fig.19 Working status of pipe network components after 50 years of service

551



552

553 Fig.20 Failure probability of most corroded and least corroded pipes in 50 years

554 6.2 System reliability analysis

555 According to the results in Section 6.1, each component of the pipeline system has different
556 time-dependent reliability and is independent of each other. Therefore, the survival signature

theory under the INID assumption was used to analyze the reliability of the system. To better analyze system reliability, the mainstream network system in Fig.21 is defined. The red demanding node indicates that water needs to be delivered to the target to fulfill the user's needs. And the blue junction node represents the pipe connector, which is used to indicate the connection point of the pipes. Our goal is to send water to all the demanding nodes.

As can be seen from Fig. 21, the pipeline of the Tsinghua campus is a double-ring transportation system (represented by the purple line and the red line), and the two-ring transportation lines are connected by pipelines in green. These pipes transport water to the corresponding location, which are referred to as mainline pipelines. From there, branch pipelines (black line) transport the water to demand points or each level of the building. As a result, the failure of branch pipes will result in the water being shut off at certain floors or demand areas of the building, but the failure of the combination of mainline pipes would result in the buildings in a certain region having their collective water supply turned off. This essay focuses on the breakdown of the main pipeline system, which caused the water to be turned off to local structures. Figure 22a is a simplified model of the mainstream pipelines in Fig.21, where each line represents pipes connected in series, and the color of the lines in Fig.22a corresponds to those in Fig.21, representing pipelines in different areas. Figure 22b-22i show the eight system failure modes for regional water cut-off, where the black line indicates that there is no water supply for demanding nodes in this region.

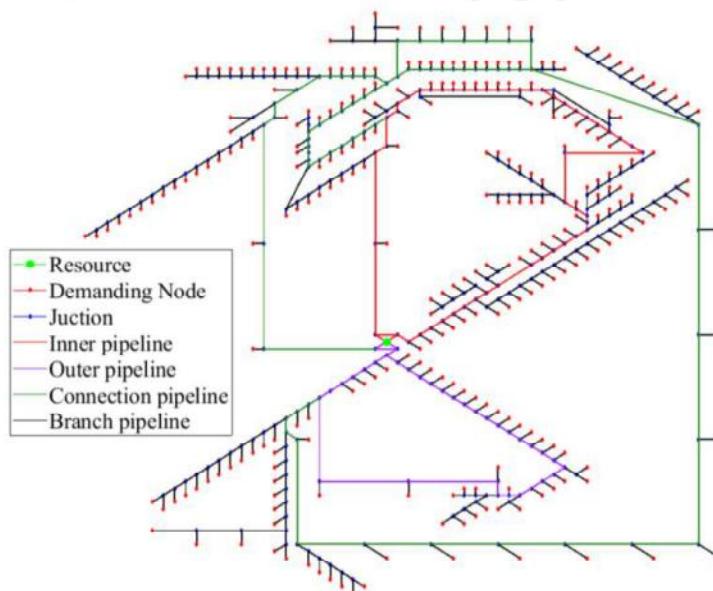
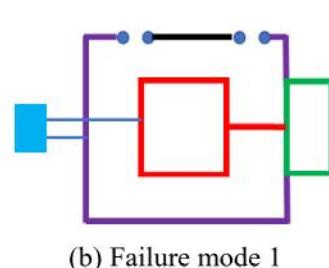
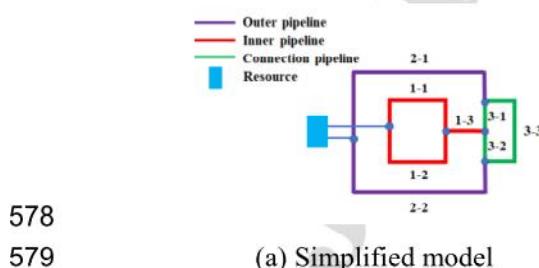


Fig.21 Components in the mainstream pipeline network



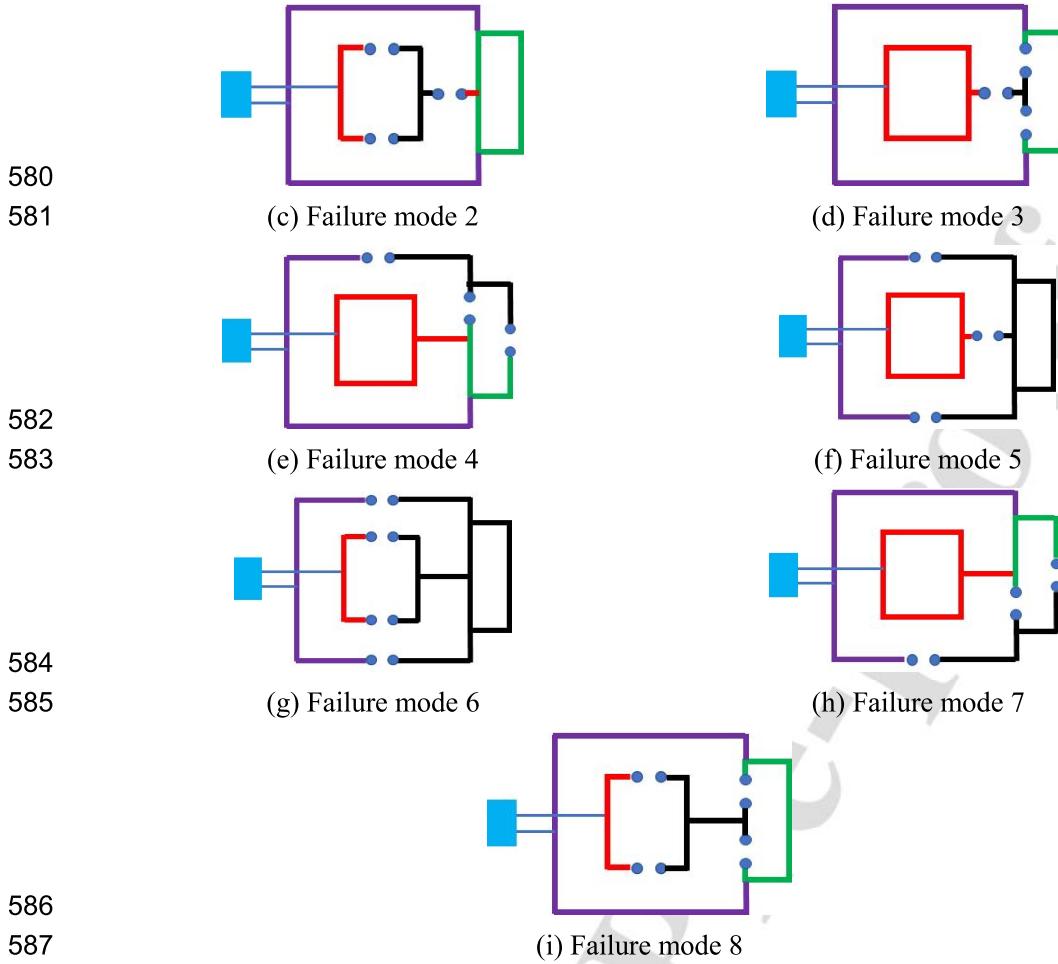


Fig.22 Simplified model of the mainstream system and system failure modes

The mainstream pipeline system has a total of 261 components. In Fig. 22a, subsystem 1 has 93 components, of which 17 are in subsystem 1-1, 72 are in subsystem 1-2, and 4 are in subsystem 1-3. 110 components make up subsystem 2, of which 67 components make up subsystem 2-2 and 43 components make up subsystem 2-1. Moreover, there are 58 components in subsystem 3, including 12 in subsystem 3-1, 8 in subsystem 3-2, and 38 in subsystem 3-3. The flow chart in Fig. 3 is used to determine the three subsystems' minimal divide-and-conquer units. Subsystem 1 is divided into 16 units, 3 units each containing 5 components, and 13 units containing 6 components. Subsystem 2 is divided into 16 units, 2 units each containing 6 components, and 14 units containing 7 components. Subsystem 3 is divided into 8 units, 6 units each containing 7 components, and 2 units containing 8 components. The survival signature under the i.i.d assumption recorded in Table.3 is calculated by generating all the state vectors when $l_k (k = 1,2,3)$ components working in subsystem k , then calculate the number of state vectors that conform to the failure modes in Fig. 22. For example, when $l_1 = 93, l_2 = 110, l_3 = 56$, there are $C_{58}^{56} = 1653$ state vectors and only $C_{12}^{10} + C_8^6 + C_{38}^{36} = 797$ states vectors lead to system failure (failure mode Fig.22b in subsystem 3-1, 3-2 and 3-3). And the survival signature

$$\Phi(l_1 = 93, l_2 = 110, l_3 = 56) = 1 - \frac{797}{1653} = 0.5178.$$

Figure 23 shows the values of survival signatures $\Phi(92,110,56), \Phi(91,110,56)$ obtained from **Algorithm 4** and the time-dependent reliability of each component. When the survival

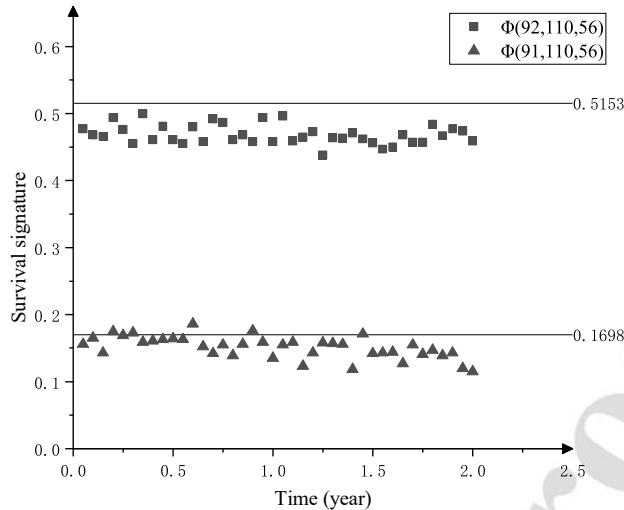
signature is 0 or 1 in the i.i.d assumption, it is still 0 or 1 while components are INID distributed. The survival signatures calculated by **Algorithm 4** in Fig.23 are significantly different from that calculated under the i.i.d assumption. Both $\Phi(92,110,56)$ and $\Phi(91,110,56)$ under component INID assumption are different from those under i.i.d assumption and fluctuate over time. The average distance between the results of $\Phi(92,110,56)$ calculated under two assumptions is 0.05126, while the distance of $\Phi(91,110,56)$ is 0.0250. **Algorithm 1** and **Algorithm 4** (sample number for each survival signature is 10000) are then combined to calculate the system reliability. For comparison, the components are set as different types and use Eq. (3) to calculate the analytical solution of system reliability. The results are shown in Fig. 23 and Table.4. The reliability of big systems calculated using the suggested technique is proven to be highly accurate as demonstrated in Table 4.

618

619 Table.3 - Survival signature of pipeline system under i.i.d assumption

l_1	l_2	l_3	Survival $\Phi(l_1, l_2, l_3)$	signature
93	110	58		1.0000
93	110	57		1.0000
93	110	56		0.5178
93	110	55		0.1182
93	109	58		1.0000
93	109	57		1.0000
93	109	56		0.2980
93	109	55		0.0000
93	108	58		0.4806
93	108	57		0.4806
93	108	56		0.0000
93	108	55		0.0000
93	≤ 107	≤ 54		0.0000
92	110	58		1.0000
92	110	57		1.0000
92	110	56		0.5153
92	110	55		0.1131
92	109	58		1.0000
92	109	57		1.0000
92	109	56		0.2852
92	109	55		0.0000
92	108	58		0.4599
92	108	57		0.4599
92	108	56		0.0000
92	108	55		0.0000
92	≤ 107	≤ 54		0.0000
91	110	58		0.3693
91	110	57		0.3693
91	110	56		0.1698
91	110	55		0.0000
91	109	58		0.3693
91	109	57		0.3693
91	109	56		0.0000
91	109	55		0.0000
≤ 90	≤ 108	≤ 54		0.0000

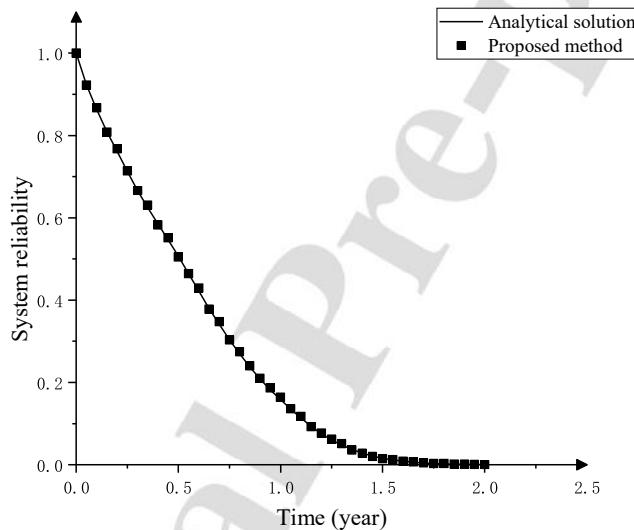
620



621

622

Fig.23 Survival signatures for pipeline system in Tsinghua campus



623

624

Fig.24 System reliability for the pipeline in the Tsinghua campus

625

626

627

628

629

630

631

632

633

634

635

636

637

Table 5 displays the computation time to determine the reliability of the water supply network system for the Tsinghua campus using four alternative approaches in order to demonstrate the computation efficiency of the suggested technique (the result for each method is the average time for ten calculation time points). Method 1 and method 2 are the proposed method and analytical method by setting components as different types using Eq. (3). Method 3 still uses the proposed **Algorithm 1** and **Algorithm 4**, but the water supply system is regarded as a whole rather than divided into three subsystems. In this way, the survival signature becomes system structure and according to the failure modes in Fig. 22, five kinds of system signatures need to be computed ($\Phi(261)$, $\Phi(260)$, $\Phi(259)$, $\Phi(258)$, $\Phi(257)$). Method 4 also obtains the analytical solution by the BDD-based method. It first builds the BDD structure of the system and then computes the probability of all Boolean variables pointing to the 1-node based on the Boolean function.

Table.4 - The system reliability under different years solved by the two methods

Time (year)	System reliability (Analytical solution)	System reliability (Proposed method)
0	1	1(0%)
0.05	0.92732	0.9224(0.0867%)
0.1	0.86350	0.8670(0.4053%)
0.15	0.80928	0.8079(0.1705%)
0.2	0.76143	0.7674(0.7841%)
0.25	0.71271	0.7142(0.2091%)
0.3	0.66533	0.6660(0.1007%)
0.35	0.62551	0.6309(0.8617%)
0.4	0.58576	0.5821(0.6248%)
0.45	0.54670	0.5513(0.8414%)
0.5	0.50691	0.5045(0.4754%)
0.55	0.46478	0.4643(0.1033%)
0.6	0.42188	0.4294(1.7825%)
0.65	0.37991	0.3778(0.5554%)
0.7	0.34165	0.3475(1.7123%)
0.75	0.30312	0.3032(0.0264%)
0.8	0.26956	0.2745(1.8326%)
0.85	0.23707	0.2400(1.2359%)
0.9	0.20786	0.2099(0.9814%)

638

639 Table.5 – Comparison of average calculation time at each time point for four methods

	Proposed method $K = 3$	Analytical method	Proposed method $K = 1$	BDD-based method
Calculation time (sec)	41.5479	58.321447	0.6050	548.321447

640

641 It can be found from Table 5 that the calculation time in method 3 is the shortest, followed
 642 by method 1, which indicates that the proposed method can simplify the computation and define
 643 the number of subsystems flexibly. The computation time of the proposed method is largely
 644 dependent on the number of survival signatures $\Phi(l_1, \dots, l_K)$. The calculation of each survival
 645 signature requires a separate weighted random sampling of all subsystems, which is the main
 646 operational consumption of the proposed method. According to Table 3, there are 23 survival
 647 signatures $\Phi(l_1, l_2, l_3)$ that need to be computed when the system is divided into 3 subsystems.
 648 However, only 5 system signatures $\Phi(l)$ need to be analyzed in method 3 when the number of
 649 subsystems $K = 1$. Therefore, method 3 has a significantly reduced computation time
 650 compared to method 1. As for method 2, there are 261 types of components and 2^{261}
 651 $\Phi(l_1, \dots, l_{261})$. Although most of $\Phi(l_1, \dots, l_{261})$ are 0 and there are 35172068 state vectors \underline{x}
 652 with $\phi(\underline{x}) = 1$, so the probability structure for survival signatures should be calculated
 653 35172068 times at each time point, which will take a lot of time. The calculation time of system
 654 reliability based on the BDD method is greatly reduced, but the process of constructing the
 655 BDD structure is very time-consuming, which requires 519sec to build the BDD structure with
 656 3999602 nodes in this case. In general, the proposed method has a great improvement in
 657 computational efficiency.

658 6.3 Component importance index analysis

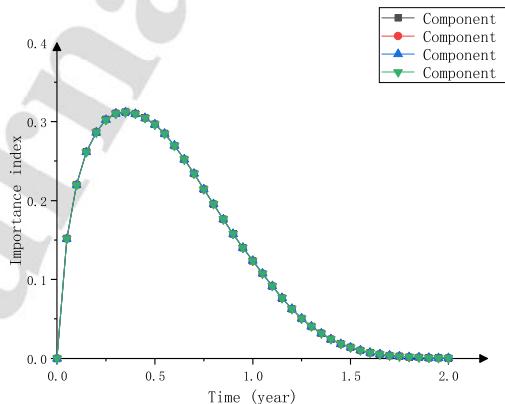
659 To better measure the importance of components in the system, the pipeline system is
 660 further subdivided into eight sections as shown in Fig.22a (subsystem 1-1, subsystem 1-2,
 661 subsystem 1-3, subsystem 2-1, subsystem 2-2, subsystem 3-1, subsystem 3-2 and subsystem 3-

662 3). In this way, each subsystem is a series structure. The failure of any component will cause
 663 the failure of the series system. Based on the developed approach in Section 5, the failure
 664 probability of each component is set as 0 and 1 respectively. The divide-and-conquer algorithm
 665 and the system structure simulation method are utilized to obtain the importance index of the
 666 component. To better show the importance index of each component in the subsystem, take the
 667 subsystem 1-3 as an example, there are only four components in the subsystem 1-3, and
 668 components in the subsystem are connected in series. The importance indexes of the four
 669 components are shown in Fig.26. Figure 27 shows the relative importance index values of the
 670 most critical component in each subsystem over time.

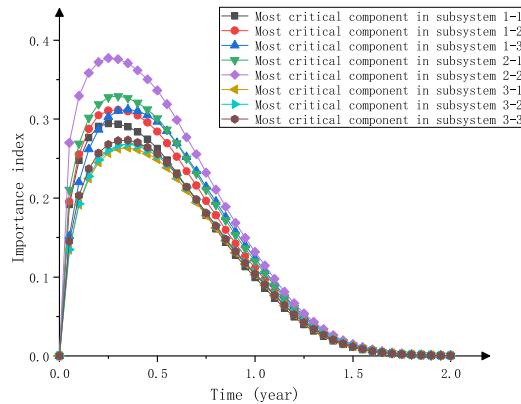
671 As shown in Fig.25, there is little difference in the importance index of components in the
 672 same subsystem. Since for series systems, $P(\phi(X) = 1|X_i = 0)$ is equal to 0 for each
 673 component i in the system, and the failure of a single component will lead to system failure.
 674 But for pipeline systems, no matter what the states are for components in other subsystems,
 675 when one component is failure in a series subsystem, the rest components in the subsystem
 676 should be in working to make the system work. Hence, $P(\phi(X) = 1|X_i = 0)$ is only related to

677 $\prod_{j \neq i}^{m_k} P_{f_j}$. Also, $P(\phi(X) = 1|X_i = 1)$ of component i in the same subsystem is directly
 678 connected to the failure probability P_{f_i} . However, it should be realized that if the system has a
 679 large number of components, the change in the reliability of a single component has little impact
 680 on the whole system, resulting in a small difference in the importance index of the components
 681 in the series structure.

682 Similarly, since the eight subsystems are in a series structure, the importance index of the
 683 most critical component in each subsystem as shown in Fig.26 could reflect the importance of
 684 the corresponding subsystem. According to the findings in Fig. 26, subsystem 2-2 has the
 685 greatest component importance index, which has a significant impact on the pipeline system's
 686 overall reliability. This is mostly due to subsystem 2-2 having the most components, each of
 687 which has a rather significant failure chance. Each subsystem's maintenance resources are
 688 allocated based on the relative magnitude of the significance index, which can increase system
 689 reliability and save costs.



690
 691 Fig.25 Component importance index in subsystem 1-3



692
693 Fig.26 Importance index of the most critical component in each subsystem
694

695 7. Conclusion

696 In this paper, a reliability approach is developed to analyze systems with independent but
697 not identical components based on survival signature theory. A combination of the divide-and-
698 conquer algorithm and system failure modes is employed to derive the corresponding system
699 equations of various state vectors and calculate the value of the structure function. The findings
700 demonstrated that the divide-and-conquer algorithm is more effective than the conventional
701 exhaustive algorithm at calculating probability structures, particularly when the total number
702 of components exceeds 1000. In this case, the calculation time can be reduced by about 1000
703 times when compared to the conventional approach. The investigation also shows that the size
704 of the minimum divide-and-conquer unit is the main factor that affects the efficiency of the
705 algorithm. In system structure calculation, the weighted random sampling has higher
706 computation efficiency, especially for systems with small failure probability and a large number
707 of components. Based on an example analysis, it is shown the calculated survival signatures are
708 no longer a constant value under INID components. The proposed method shows good
709 agreement with the analytical solution in the calculation of system reliability. Compared with
710 other methods (BDD-based method and tradition survival signature theory), the proposed
711 method has more advantages in system division and reliability calculation. It was also shown
712 the developed method can effectively calculate the importance index of components by setting
713 the failure probability of components as 0 or 1.

714 The purpose of this study is to apply the independently distributed condition to the survival
715 signature theory under independently and identically distributed conditions. The primary
716 solution procedure is split into two steps: solving the system's survival signatures and each
717 survival signature's accompanying probability structure. When the components are non-
718 identically distributed, it also means $K = m$, which is a general system reliability problem. The
719 survival signature method is a new attempt to calculate the reliability of large complex systems.
720 For example, it can analyze the influence of the different number of failure components in each
721 subsystem on system reliability and select the most important components in the system for
722 maintenance and preventive measures. It is also worth noting that when $K = m$, the subsystem
723 definition in the survival signature theory can be more flexible, rather than based on the failure
724 probability distribution type of each component. For example, the water supply network can be
725 divided according to the pipeline transportation area, and the automobile engine subsystem can

726 be divided according to the function of each component.

727 In addition, this work can be extended to component-dependent systems. The only
 728 difference between dependent components and independent components is the probability
 729 structure. When components are independent of each other, the joint distribution is the product
 730 of the failure probability of each component, while it can be replaced by copula functions when
 731 considering component dependence. Therefore, the application of this method in a component-
 732 correlated system can be considered in future work.

733 Nevertheless, because the survival signature under INID is dependent on the component
 734 failure probabilities and must be calculated at several points in time, the signature theory lacks
 735 the capacity to distinguish between system structure and failure time. This is another flaw and
 736 restriction of this work, despite the fact that the suggested approach is more effective than basic
 737 Monte Carlo.

738

739 Appendix A. -Pipeline information and stress calculation formula

740 Detailed information on the investigated pipeline is given in the following table and figure.

741 Table A.1. Pipeline information

Variable	Units	Mean	Standard deviation	References
Hydraulic pressure P	Mpa	From Epanet	-	-
Inner diameter D	mm	From BIM	-	-
Wall thickness d	mm	From BIM	-	-
Bending moment coefficient K_m	-	0.235	0.04	Sadiq [46]
Calculation coefficient C_d	-	1.32	0.20	Sadiq [46]
Modulus of elasticity of pipe E_p	MPa	From BIM	-	-
Deflection coefficient K_d	-	0.108	0.0216	Sadiq [46]
Impact factor I_c	-	1.5	0.375	Sadiq [46]
Surface load coefficient C_t	-	0.12	0.024	Sadiq [46]
Wheel load F	N	412,000	20,000	JTG D60-2015 [47]
Pipe effective length A	mm	-	-	JTG D60-2015 [47] GB50268-2008[48]
Soil unit weight γ	N/mm ³	18.85×10^{-6}	18.85×10^{-7}	Sadiq [46]
Poisson's ratio of pipe material ν_p	-	0.24	0.06	-
Fracture toughness K_{IC}	N/mm ^{1.5}	21.88	-	Fracture toughness test of ductile iron [49]
Multiplying constant – Internal K	-	0.92	0.18	Marshall [45]
Multiplying constant – External K	-	2.54	0.5	Marshall [45]
Exponential constant – Internal n	-	0.4	0.08	Marshall [45]
Exponential constant – External n	-	0.32	0.06	Marshall [45]

742 The pipeline working environment information and material performance information, such as
 743 wheel load F , soil unit weight γ , bending moment coefficient K_m , Impact factor I_c , are obtained

744 from literature and standards in the form of random variables. The attribute information of each pipe
 745 can be directly extracted from the BIM model and substituted into the corresponding model for
 746 calculation. The working water pressure of the pipeline is obtained by hydraulic analysis after
 747 establishing an equal-scale model in Epanet software.

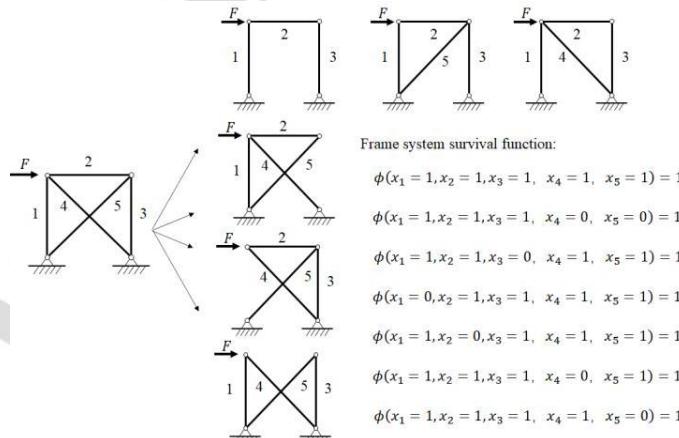
748 The formula for calculating the external load pressure on pipelines is shown in the following
 749 table, and the values of each variable in the formula can be obtained from Table A.1.

750 Table A.2. Stress calculation formula

Stress type	Model	Reference
σ_F , hoop stress due to internal fluid pressure	$\frac{pD}{2d}$	Rajani [50]
σ_S , soil pressure	$\frac{3K_m\gamma B_d^2 C_d E_p D^2}{E_p d^3 + 3K_d p D^3}$	Ahammed and Melchers [51]
σ_V , traffic pressure	$\frac{3K_m I_c F C_t E_p D d}{A(E_p d^3 + 3K_d p D^3)}$	Ahammed and Melchers [51]
σ_P , axial stress due to internal fluid pressure	$\frac{p}{2} \left(\frac{D}{d} - 1 \right) v_p$	Rajani [50]
σ_h , total hoop stress	$\sigma_h = \sigma_F + \sigma_S + \sigma_V$	Rajani [50]
σ_a , total axial stress	$\sigma_a = \sigma_P + (\sigma_F + \sigma_S + \sigma_V) v_p$	Rajani [50]

751 Appendix B. - Application scope of the proposed methods and open source

752 In this part, the applicability of the proposed method is explained. Firstly, the codes for the reliability
 753 analysis of the series-parallel system and Tsinghua water supply network can be obtained on GitHub
 754 (<http://github.com/JoyZheng2022/Codes-for-paper>). Secondly, this paper mainly analyzes the
 755 calculation method of system reliability under the INID assumption. Components don't need to be
 756 independent of each other. The algorithms are realized mainly by the occurrence probability of
 757 each state vector ($P(x)$ in Eq. (10)). Therefore, combined with copula theory, the proposed
 758 method can also be applied in multi-state systems [28] and dependent-component systems
 759 [19],[52]. Finally, **Algorithm 4** is operated based on system failure modes, which allows the
 760 proposed algorithm to be applied to more systems (including engineering mechanics structure
 761 analysis) than just graph-based systems. For instance, in the structural analysis of the following
 762 frame, when the frame is subjected to lateral forces, seven state vectors make the system
 763 function. Then by calculating the probability of each state vector and substituting it into Eq.
 764 (10), the reliability of the frame under lateral force can still be calculated by the proposed
 765 method.



766
 767 Fig.B-1 System failure modes for a typical frame

768 **Acknowledgments**

769 The authors gratefully acknowledge the financial support from National Natural Science
 770 Foundation of China under project number of Grand No. 51908324&52111540161. The support
 771 from Tsinghua University Initiative Scientific Research Program (20213080003) is also greatly
 772 appreciated.

773

774 **References**

- 775 [1]. An analytical framework for local and global system kinematic reliability sensitivity of robotic
 776 manipulators. *Applied Mathematical Modelling*, 102, 331-350.
 777 doi:<https://doi.org/10.1016/j.apm.2021.09.021>
- 778 [2]. Reliability modeling for a repairable (k_1, k_2) -out-of- n : G system with phase-type vacation time. *Applied Mathematical Modelling*, 91, 311-321. doi:<https://doi.org/10.1016/j.apm.2020.08.071>
- 780 [3]. Matsuoka, T. . (2021). Procedure to solve mutually dependent fault trees (ft with loops). *Reliability Engineering and System Safety*, 214.
- 782 [4]. Pliego Marugan, A. , Garcia Marquez, F. P. , & Lev, B. . (2017). Optimal decision-making via
 783 binary decision diagrams for investments under a risky environment. *International Journal of
 784 Production Research*, 55(17-18), 1-16.
- 785 [5]. Bryant RE. Binary decision diagrams. *Handbook of model checking*. Cham: Springer; 2018. p.
 786 191–217.
- 787 [6]. Alkaff, A. (2022). State space and binary decision diagram models for discrete standby systems
 788 with multistate components. *Applied Mathematical Modelling*, 110, 298-319.
 789 doi:<https://doi.org/10.1016/j.apm.2022.05.045>
- 790 [7]. Jia, H. , Ding, Y. , Peng, R. , Liu, H. , & Song, Y. . (2020). Reliability assessment and
 791 activation sequence optimization of non-repairable multi-state generation systems considering
 792 warm standby. *Reliability Engineering and System Safety*, 195.
- 793 [8]. Bouhamed, H. , Masmoudi, A. , Lecroq, T. , & Reba? A. . (2015). Structure space of bayesian
 794 networks is dramatically reduced by subdividing it in sub-networks. *Journal of Computational
 795 & Applied Mathematics*, 287, 48-62.
- 796 [9]. Zarei, E. , Azadeh, A. , Khakzad, N. , Aliabadi, M. M. , & Mohammadfam, I. . (2017).
 797 Dynamic safety assessment of natural gas stations using bayesian network. *Journal of
 798 Hazardous Materials*, 321(jan.5), 830-840.
- 799 [10].M. Kvassay, E. Zaitseva, V. Levashenko, and J. Kostolny, "Binary decision diagrams in
 800 reliability analysis of standard system structures," in 2016 International Conference on
 801 Information and Digital Technologies (IDT), 2016, pp. 164-172.
- 802 [11].Coolen, F. , & Coolen-Maturi, T. . (2013). Generalizing the signature to systems with multiple
 803 types of components. *Advances in Intelligent & Soft Computing*, 170, 115-130.
- 804 [12].Aslett, L.J.M., Coolen, F.P.A. and Wilson, S.P. (2015), Bayesian Inference for Reliability of
 805 Systems and Networks Using the Survival Signature. *Risk Analysis*, 35: 1640-1651.
 806 <https://doi.org/10.1111/risa.12228>
- 807 [13].Feng, G. , Patelli, E. , Beer, M. , & Coolen, F. . (2016). Imprecise system reliability and
 808 component importance based on survival signature. *Reliability Engineering & System Safety*,
 809 150, 116-125.
- 810 [14].George-Williams, H., Feng, G., Coolen, F. P. A., Beer, M., & Patelli, E. (2019). Extending the
 811 survival signature paradigm to complex systems with non-repairable dependent failures.

- 812 Proceedings of the Institution of Mechanical Engineers Part O-Journal of Risk and Reliability,
 813 233(4), 505-519. doi:10.1177/1748006x18808085
- 814 [15].Huang, X. , Coolen, F. , & Coolen-Maturi, T. . (2019). A heuristic survival signature based
 815 approach for reliability-redundancy allocation. Reliability Engineering & System Safety,
 816 185(may), 511-517.
- 817 [16].Samaniego, F. J. , & Navarro, J. . (2016). On comparing coherent systems with heterogeneous
 818 components. Advances in Applied Probability, 48(01), 88-111.
- 819 [17].Mi, J. , Beer, M. , Li, Y. F. , Broggi, M. , & Cheng, Y. . (2020). Reliability and importance
 820 analysis of uncertain system with common cause failures based on survival signature.
 821 Reliability Engineering & System Safety, 106988.
- 822 [18].Geng, F. . (2018). Survival signature-based reliability approach for complex systems
 823 susceptible to common cause failures. International Journal of Reliability, Quality and Safety
 824 Engineering, 26(4).
- 825 [19].Coolen-Maturi, T. , Coolen, F. , & Balakrishnan, N. . (2021). The joint survival signature of
 826 coherent systems with shared components. Reliability Engineering & System Safety, 207,
 827 107350-.
- 828 [20].Patelli, E. , Feng, G. , Coolen, F. , & Coolen-Maturi, T. . (2017). Simulation methods for
 829 system reliability using the survival signature. Reliability Engineering & System Safety, 167,
 830 327-337.
- 831 [21].Liang, H. , Mi, J. , Bai, L. , & Cheng, Y. . (2021). Reliability analysis of discrete multi-state
 832 systems based on survival signature. IOP Conference Series: Materials Science and
 833 Engineering, 1043(5), 052029 (8pp).
- 834 [22].Qin, J., & Coolen, F. P. A. (2022). Survival signature for reliability evaluation of a multi-state
 835 system with multi-state components. Reliability Engineering & System Safety, 218.
 836 doi:10.1016/j.ress.2021.108129.
- 837 [23].Li, Y. , Coolen, F. , Zhu, C. , & Tan, J. . (2020). Reliability assessment of the hydraulic system
 838 of wind turbines based on load-sharing using survival signature. Renewable Energy, 153, 766-
 839 776.
- 840 [24].Bai, X. , Li, X. , Balakrishnan, N. , & He, M. . (2021). Statistical inference for dependent
 841 stress-strength reliability of multi-state system using generalized survival signature. Journal of
 842 Computational and Applied Mathematics, 390(2), 113316.
- 843 [25].Tavangar, M., & Hashemi, M. (2022). Reliability and maintenance analysis of coherent
 844 systems subject to aging and environmental shocks. Reliability Engineering & System Safety,
 845 218, 108170. doi:<https://doi.org/10.1016/j.ress.2021.108170>
- 846 [26].Liu, Y. , Shi, Y. , Bai, X. , & Liu, B. . (2018). Stress-strength reliability analysis of system
 847 with multiple types of components using survival signature. Journal of Computational and
 848 Applied Mathematics, 375-398.
- 849 [27].Liu, Yiming, Shi, Yimin, Bai, & Xuchao, et al. (2018). Stress strength reliability analysis of
 850 multi-state system based on generalized survival signature. Journal of computational and
 851 applied mathematics.
- 852 [28].Bai, X., Zhang, J., He, M., & Balakrishnan, N. (2023). Inference for stress-strength reliability
 853 of multi-state system with dependent stresses and strengths using improved generalized
 854 survival signature. JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS,
 855 420, 114809. doi:<https://doi.org/10.1016/j.cam.2022.114809>

- 856 [29]. Aslett, L. . (2013). Reliabilitytheory: tools for structural reliability analysis.
- 857 [30]. Behrensdorf, J. , Regenhardt, T. E. , Broggi, M. , & Beer, M. . (2021). Numerically efficient
858 computation of the survival signature for the reliability analysis of large networks. Reliability
859 Engineering & System Safety, 216, 107935.
- 860 [31]. Feng G, Reed S, Patelli E, Beer M, Coolen FP. Efficient reliability and uncertainty assessment
861 on lifeline networks and imprecise survival signature. 2nd ECCOMAS thematic conference on
862 uncertainty quantification in computational sciences and engineering; 2017.
- 863 [32]. Reed, S. . (2017). An efficient algorithm for exact computation of system and survival
864 signatures using binary decision diagrams. Reliability Engineering & System Safety, 165, 257-
865 267.
- 866 [33]. Reed, S. , Lofstrand, M. , & Andrews, J. . (2019). An efficient algorithm for computing exact
867 system and survival signatures of k-terminal network reliability. Reliability Engineering &
868 System Safety, 185(may), 429-439.
- 869 [34]. Yi, H. , Cui, L. , & Balakrishnan, N. . (2021). Computation of survival signatures for multi-
870 state consecutive-k systems. Reliability Engineering & System Safety, 208, 107429.
- 871 [35]. Kleinberg, & Tardos, Eva. (2014). Algorithm design (First edition.; International ed.).
- 872 [36]. Efraimidis, P. , & Spirakis, P. . (2008). Weighted random sampling. Springer New York.
- 873 [37]. Efraimidis, P. S. , & Spirakis, P. G. . (2006). Weighted random sampling with a reservoir.
874 Information Processing Letters, 97(5), 181-185.
- 875 [38]. Zhang, K., Chen, N., Zeng, P., Liu, J., & Beer, M. (2022). An efficient reliability analysis
876 method for structures with hybrid time-dependent uncertainty. Reliability Engineering &
877 System Safety, 228, 108794. doi:<https://doi.org/10.1016/j.ress.2022.108794>
- 878 [39]. Hardy, G. , Lucet, C. , & Limnios, N. . (2007). K-terminal network reliability measures with
879 binary decision diagrams. IEEE Transactions on Reliability, 56(3), 506-515.
- 880 [40]. Ramirez-Marquez, J. E. , & Coit, D. W. . (2005). Composite importance measures for multi-
881 state systems with multi-state components. Reliability IEEE Transactions on, 54(3), 517-529.
- 882 [41]. Rossman, L. A. . (2000). Epanet 2 users manual.
- 883 [42]. Li, C. Q., and Mahmoodian, M. (2013). “Risk based service life prediction of underground cast
884 iron pipes subjected to corrosion.” J. Reliab. Eng. Syst. Saf., 119(Nov), 102–108.
- 885 [43]. Li, C. Q., and Mahmoodian, M. (2018). Reliability-based service life prediction of corrosion-
886 affected cast iron pipes considering multifailure modes. Journal of Infrastructure Systems.
- 887 [44]. Laham, A. S. (1999). Stress intensity factor and limit load handbook, British Energy
888 Generation, Barnwood, U.K.
- 889 [45]. Marshall, P. (2001). The residual structural properties of cast iron pipes—Structural and design
890 criteria for linings for water mains, U.K. Water Industry Research, Westminster, U.K.
- 891 [46]. Sadiq, R., Rajani, B., & Kleiner, Y. (2004). Probabilistic risk analyis of corrosion associated
892 failures in cast iron water mines. Reliability Engineering & System Safety, 86, 1-10.
893 doi:[10.1016/j.ress.2003.12.00](https://doi.org/10.1016/j.ress.2003.12.00).
- 894 [47]. JTG D60-2015. (2015). “General Specifications for design of Highway Bridges and Culverts.”
895 Industrial standard of the People’s Republic of China, Beijing.
- 896 [48]. GB50268-2008. (2008). “Code for construction and acceptance of water supply and drainage
897 pipeline projects.” National standards of the People’s Republic of China, Beijing.
- 898 [49]. Fracture toughness test of ductile iron [J]. International rolling stock technology, 1977(01):29-
899 45.

- 900 [50]. Rajani, B., et al. (2000). Investigation of grey cast iron water mains to develop a methodology
901 for estimating service life, American Water Works Association Research Foundation, Denver.
- 902 [51]. Ahammed, M., and Melchers, R. E. (1994). "Reliability of underground pipelines subject to
903 corrosion." J. Transp. Eng., 10.1061/(ASCE) 0733-947X(1994)120:6(989), 989–1002.
- 904 [52]. Behrensdorf, J., Broggi, M., & Beer, M. (2019). Reliability Analysis of Networks
905 Interconnected With Copulas. ASCE-ASME JOURNAL OF RISK AND UNCERTAINTY IN
906 ENGINEERING SYSTEMS PART B-MECHANICAL ENGINEERING, 5(4).
907 doi:10.1115/1.4044043
- 908