

Chapter 10

Network Invariants and Their Use in Performability Analysis



Ilya Gertsbakh and Yoseph Shpungin

Abstract Network-type systems with binary components have important structural parameters known in literature as Signature, Internal Distribution, D-spectra and BIM-spectra. The knowledge of these parameters allows obtaining the probabilistic description of network behaviour in the process of their component failures, and probabilistic description of such network parameters as resilience, component importance, system failure probability as a function of component failure probability q , and the approximation to reliability if q tends to 0. When the network has many components, the exact calculation of Signatures or D-spectra becomes a very complicated issue. We suggest using efficient Monte Carlo procedures. All relevant calculations are illustrated by examples of networks, including flow in random networks and network structural comparison in the process of network gradual destruction process.

Keywords Network structure invariants · Network resilience · Shock model · Reliability approximation

10.1 Introduction

Reliability and availability [1] are two important probabilistic attributes of performability of any network or system with binary states. This chapter lays down discussion of important structural parameters known as Signature, Internal Distribution, D-spectra and BIM-spectra. Before we do that some definitions are provided to make the discussion meaningful.

Prof. Gertsbakh has sadly passed away prior to the publication of this manuscript.

I. Gertsbakh
Ben Gurion University, Beer Sheva, Israel

Y. Shpungin (✉)
Shamoon College of Engineering, Beer Sheva, Israel
e-mail: yfromb@gmail.com

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

K. B. Misra (ed.), *Handbook of Advanced Performability Engineering*,
https://doi.org/10.1007/978-3-030-55732-4_10

10.1.1 Networks, Node and Edge Failures. Success Criteria

We meet networks every day and everywhere in our life. For the formal study of network properties, we must operate with abstract models of networks. In further, our principal network model will be a triple $N = (V, E, T)$, where V is the vertex or node set, E is the edge or link set and T is a set of special nodes called terminals, $T \in V$. In simple words, a network is a collection of circles (nodes) and links (edges), i.e. line segments connecting the nodes. Terminals are special nodes that do not fail and they are represented as bold circles, like in Fig. 10.1.

Our exposition will be centred around network behaviour when its elements (nodes and/or links) fail. We will deal with so-called binary elements that can be in two states up and down denoted by 1 and 0, respectively. When speaking about links, link i failure means that this link is erased, i.e. it does not exist.

The state of link i , $i = 1, \dots, n$ is denoted by binary variable x_i . If $x_i = 1$, link i is up; if $x_i = 0$, link i is down. x_i is often called link indicator variable. In some models, the elements subjected to failure are network nodes (vertices). If the indicator variable of node j is $y_j = 0$, i.e. node j is down, it means that all links incident to node j are erased, but the node itself remains intact. By an agreement, the terminals do not fail.

By network state, we mean the set of all its elements (nodes and edges) that are in up state. We will distinguish network UP (operating) and DOWN (non-operating) states according to a certain criterion.

Below we give several examples of different UP and DOWN criteria. All examples relate to the network are shown in Fig. 10.1. This network has two terminals: 1 and 6.

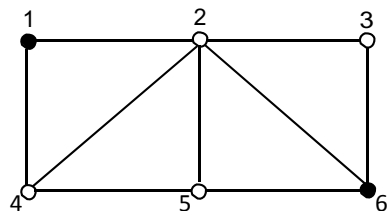
Terminal Connectivity Criterion. Nodes Unreliable, Edges Reliable

We say that the network is UP if each pair of terminals is connected by a path of non-erased elements. Let node 2 is up, and nodes 3, 4, 5 are in the down state. Then the network is UP, because the node 2 connects terminals 1 and 6. Let now nodes 2 and 4 are down, and nodes 3 and 5—up. Then the network is DOWN.

Terminal Connectivity Criterion. Edges Unreliable, Nodes Reliable

Suppose that edges (1,2), (2,5) and (5,6) are up, and all other edges are down. The network is UP. Let now the edges (1,2) and (1,4) be down, and all other edges are up. The network is DOWN.

Fig. 10.1 Network with six nodes and nine edges. Nodes 1 and 6 are Terminals



Max Component and Max Cluster Criteria. Nodes Unreliable

For this example, we need the definition of a *component*. A subset $V_1 \subset V$ is called an *isolated component* of N if all nodes of V_1 are connected to each other and there are no edges of type $e = (a, b)$, where $a \in V_1$ and $b \in V - V_1$. An isolated node is considered as an isolated component. The *size* of a component is the number of nodes in it.

We say that the network is in UP state if the maximal component has at least x nodes, where x is some given number. Suppose that $x = 3$. Let the nodes 3 and 5 are up, and the rest of the nodes—down. Then the maximal component consisting of nodes 3, 5, 6 and edges (3,6), and (5,6) is of size 3, and therefore N is UP. Let now nodes 3 and 4 are up, and nodes 2, 5 are down. Obviously N is DOWN.

By the definition, an isolated component of N is called a *cluster* if it contains at least one terminal node. We say that the network is *UP* if it contains a cluster of size at least x , where x is some given number. Let $x = 4$. If the nodes 2 and 5 are up and 3, 4 are down, then we have a cluster of size 4, and N is UP. If only nodes 3 and 5 are up then we have maximal cluster of size 3, and N is DOWN. (Recall that the terminals are always up.)

Further we will use the notions cut and min-cut. Appropriate definition follows.

Definition 1 A subset of network unreliable elements (c_1, c_2, \dots, c_k) is called a cut if the following condition is satisfied:

If all these elements are in state down, then the network is also in state DOWN.

A cut is called minimal (min-cut) if after removing any element, the new subset is no more a cut.

Consider for example, the network in Fig. 10.1 for the case when the nodes are unreliable. The subset of nodes (2, 3, 5) is cut, but not a min-cut. Indeed, if node 3 is removed, the remaining subset (2, 5) is still a cut. It is obvious that (2,5) is min-cut.

10.2 Destruction Spectrum and Network Reliability

10.2.1 D-Spectrum and CD-Spectrum

Definition 2 Let $\pi = e_{i_1}, e_{i_2}, \dots, e_{i_n}$ be a permutation of all unreliable elements (edges or nodes). Start with a network with all elements being up and ‘erase’ the elements in the order they appear in π , from left to right. Stop at the first element e_{i_r} when the network becomes DOWN. The ordinal number r of this element is called the anchor of permutation π and denoted $r(\pi)$.

Remark 1 Note that the anchor value for given π depends only on the network structure and its DOWN definition. It is completely separated from the stochastic mechanism that governs the node or edge failures in a real network destruction process.

Example 1 Consider the network shown in Fig. 10.1. In this network with two terminals 1 and 6, the edges are reliable and nodes 2, 3, 4, 5 are unreliable. Consider an arbitrary permutation π of node numbers, e.g. $\pi = (3, 5, 2, 4)$. We start the destruction process with all nodes in the *up* state. Erase one node after another in the order prescribed by π , from the left to right. The network becomes DOWN after erasing the third node, i.e. node 2. So we have $r(\pi) = 3$.

Definition 3 Let x_i be the number of permutations such that their anchor equals i . The set

$$D = \left\{ d_1 = \frac{x_1}{n!}, \dots, d_n = \frac{x_n}{n!} \right\} \tag{10.1}$$

is called the *D*-spectrum of the network.

Remark 2 ‘*D*’ in Definition 1 refers to the ‘destruction’ process of erasing network elements from left to right in the permutation π . *D*-spectrum is distribution of the anchor value, and obviously $\sum_{i=1}^n d_i = 1$. Numerically, the *D*-spectrum coincides with the so-called Signature introduced first in (Samaniego 1985, see [2]). It was proved there that if system elements fail independently and their lifetimes X_i have identical continuous distribution function $F(t)$, then the system lifetime distribution $F_S(T) = \sum_{i=1}^n d_i \cdot F_{i:n}(t)$ where $F_{i:n}(t)$ is the cumulative distribution function of the i th order statistics in random sample X_1, X_2, \dots, X_n .

Example 1 (continued) Table 10.1 shows all 24 permutations of the nodes. The nodes destruction of which caused the failure of the network are marked by asterisk. Directly from this table we get $x_1 = 0, x_2 = 8, x_3 = 10, x_4 = 6$, and *D*-spectrum of the network equals ($d_1 = 0, d_2 = 1/3, d_3 = 5/12, d_4 = 1/4$).

Definition 4 Let $y_b = \sum_{i=1}^b d_i, b = 1, 2, \dots, n$. Then the set (y_1, y_2, \dots, y_n) is called the Cumulative *D*-spectrum (CD-spectrum).

Remark 3 Like an anchor, both spectra (*D* and CD) depend only on the network structure and the definition of network DOWN state. That is, they are invariant with respect to the up/down probabilities of the elements.

Table 10.1 All permutations of nodes

Column 1	Column 2	Column 3	Column 4
2,3,4*,5	3,2,4*,5	4,2*,3,5	5,2*,3,4
2,3,5*,4	3,2,5*,4	4,2*,5,3	5,2*,4,3
2,4*,3,5	3,4,2*,5	4,3,2*,5	5,3,2*,4
2,4*,5,3	3,4,5,2*	4,3,5,2*	5,3,4,2*
2,5*,3,4	3,5,2*,4	4,5,2*,3	5,4,2*,3
2,5*,4,3	3,5,4,2*	4,5,3,2*	5,4,3,2*

The following theorem establishes an important combinatorial property of the CD-spectrum.

Theorem 1 *Let $C(i)$ be the number of cut sets of size i in the network. Then*

$$C(i) = y_i \cdot \frac{n!}{i!(n-i)!} \quad (10.2)$$

The proof of Theorem 1 can be found in the textbook [3].

Example 1 (continued)

We get the following CD-spectrum of our network: ($y_1 = 0, y_2 = 1/3, y_3 = 3/4, y_4 = 1$).

Using formula (10.2), we get: $C(1) = 0, C(2) = 2, C(3) = 3, C(4) = 1$.

10.2.2 Network Reliability and CD-Spectrum Monte Carlo

The following theorem gives an expression of network reliability using CD - spectrum.

Theorem 2 *If all $p_i = p$, then network static reliability $R(N)$ can be expressed in the following form:*

$$R(N) = 1 - \sum_{i=1}^n y_i \cdot \frac{n!q^i p^{n-i}}{i!(n-i)!} \quad (10.3)$$

It is clear that even for relatively small networks, the exact calculation of network CD-spectrum is extremely difficult. Below we present Monte Carlo algorithm for estimating the CD spectrum.

Algorithm 1: Evaluation of CD-spectrum

1. **Initialize** all a_i to be zero, $i = 1, \dots, n$.
2. **Simulate** permutation π of all elements.
3. **Find out** the anchor $r(\pi)$.
4. **Put** $a_r = a_r + 1$.
5. **Put** $r = r + 1$. **If** $r \leq n$ **GOTO** 4.
6. **Repeat** 2–5 M times.
7. **Estimate** y_i via $\hat{y}_i = \frac{a_i}{M}$.

Figure 10.2 shows a network with 32 unreliable nodes and 60 reliable edges. Nodes 4, 13, 27, 30 are terminals. Table 10.2 shows CD-spectrum for grid network with unreliable nodes and for terminal connectivity criterion. Table 10.3 shows CD-spectrum for the same network, but with unreliable edges and maximal cluster criterion ($x = 25$). Both spectra were obtained using algorithm 1 with $M = 10,000$.

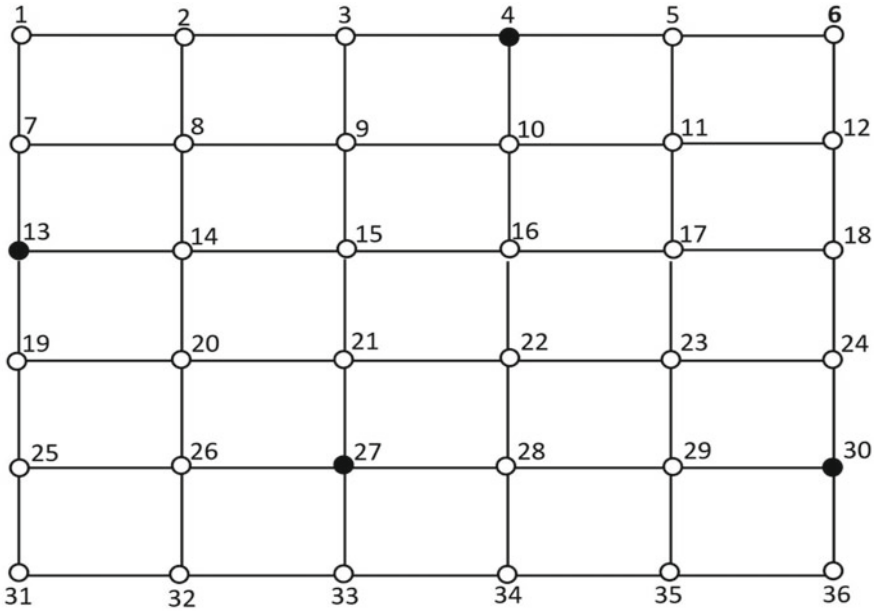


Fig. 10.2 Grid network with 32 unreliable nodes and 60 reliable edges. Nodes 4, 13, 27, 30 are terminals

Table 10.2 Grid CD-spectrum. Nodes unreliable. Terminals $T = (4, 13, 27, 30)$ terminal connectivity criterion

i	y_i	i	y_i	i	y_i	i	y_i
1	0	9	.2069	17	.9664	25	1
2	0	10	.3188	18	.9824	26	1
3	.0011	11	.4490	19	.9924	27	1
4	.0046	12	.5838	20	.9967	28	1
5	.0137	13	.7070	21	.9986	29	1
6	.0352	14	.8078	22	.9996	30	1
7	.0677	15	.8811	23	1	31	1
8	.1210	16	.9322	24	1	32	1

Remark 4 Using CD-Monte Carlo for calculating network reliability has several advantages over other methods, including the following two.

1. Since CD-spectrum is an invariant, **once estimated** it, we can calculate network reliability for any values of p .
2. Using this method, we avoid the so-called rare event phenomenon [3].

Table 10.3 Grid CD-spectrum. Edges unreliable. Terminals T = (4, 13, 27, 30). Maximal cluster criterion, $x = 25$

i	y_i	i	y_i	i	y_i	i	y_i	i	y_i
1	0	9	.0001	17	.0089	25	.2642	33	.9660
2	0	10	.0002	18	.0147	26	.3500	34	.9881
3	0	11	.0002	19	.0227	27	.4510	35	.9971
4	0	12	.0002	20	.0370	28	.5641	36	.9996
5	0	13	.0004	21	.0593	29	.6757	37	1
6	0	14	.0013	22	.0899	30	.7791	38	1
7	0	15	.0026	23	.1321	31	.8657	39	1
8	0	16	.0044	24	.1891	32	.9291	40-60	1

10.2.3 Two Alternative Methods for Evaluating Network Reliability

Method 1: Crude Monte Carlo

A common method for evaluating network reliability is the Crude Monte Carlo (CMC) method. We present below the corresponding algorithm.

Algorithm 2: CMC

1. **Set** $Y = 0$
2. **For** each element i , **Simulate** its state with probability p_i
3. **Check** the network state in accordance with given criterion
4. **If** the network state is UP **Then** $Y := Y + 1$
5. **Repeat** steps 2, 3, 4 M times.
6. **Estimate** \hat{R} as $\hat{R} = \frac{Y}{M}$

In many cases, using CMC gives good results, but unlike the CD-Monte Carlo, it has some drawbacks, including the following two.

1. For each p value, it is necessary to restart the simulation process.
2. The main disadvantage of CMC is the presence of a rare event phenomenon. That is, if $p \rightarrow 1$, then the relative error $r.e.(CMC) \rightarrow \infty$. Therefore CMC is not suitable for evaluating very reliable networks, which is actually an important practical case.

Method 2: Burtin–Pittel Approximation

Burtin–Pittel approximation provides rather accurate network reliability estimates for the case of a highly reliable network and independent and equal element unreliability $q_i = q$.

Assume that $q \rightarrow 0$, that is the network is highly reliable. Let the number of min-cuts of a minimal size r is equal to s .

Table 10.4 Grid network reliability by CMC, Destruction Monte Carlo (DMC) and Burtin–Pittel approximation (B–P)

p /Algorithm	CMC	DMC	B–P
0.7	0.6786	0.678619	0.865
0.8	0.9126	0.909937	0.96
0.9	0.9922	0.991761	0.995
0.95	0.9995	0.999185	0.999375
0.99	1	0.9999944	0.999995

Then by Burtin-Pittel approximation

$$Q(N) = 1 - R(N) \approx s \cdot q^r. \tag{10.4}$$

Note that this approximation was first suggested in a more general form by Burtin and Pittel (see [4]).

We explain this approximation using the example of the network in Fig. 10.1. As we have seen in the above example, the unreliability of this network is $Q(N) = 1 - R(N) = 2q^2 p^2 + 3q^3 p + q^4$. The main term here (when $q \rightarrow 0$) is $2q^2 p^2$, where 2 is the number of min-cuts of minimal size. Clearly that $2q^2 p^2 = 2q^2(1 + o(1))$ as $q \rightarrow 0$.

Consider now the grid network in Fig. 10.2. Obviously, the minimal size of the min-cuts is 3. All min-cuts of size 3 are as follows: (3,5,10), (7,14,19), (24,29,36), (24,29,34), (24,29,35). So in this case, we get $1 - R(N) \approx 5 \cdot q^3$.

Table 10.4 presents grid network reliability for different values of p , calculated using CMC, Destruction Monte Carlo (DMC), both with $M = 10,000$, and also Burtin-Pittel approximation (B-P). Comparing CMC and DMC we see a good correspondence up to $p = 0.95$. However, starting from $p = 0.99$ reliability values obtained using CMC with $M = 10,000$ will be 1.

Note that here we see the rare event phenomenon.

For example, we want to estimate the reliability of the order of 0.99999 with relative error at least 10%. (Note that $\text{rel. arr. (CMC)} = \frac{\sqrt{R}}{\sqrt{M \cdot (1-R)}}$.) Then we get $M = 10,000,000$.

A more detailed comparison of these methods can be found in [5].

As for B-P method, we see that it gives good approximation starting from $p = 0.9$.

10.3 Network Resilience

One of the important concepts in the analysis of the network behaviour under random attack on its elements is *network resilience*.

Definition 5 *Probabilistic resilience* [6] Assume that network element failures appear in *random order*, i.e. all $n!$ orderings are equally probable.

Let N be a network with n elements. The probabilistic resilience $\text{res}_{\text{pr}}(N; \beta)$ is the largest number of element failures such that N is still UP with probability $1 - \beta$. Formally,

$$\text{res}_{\text{pr}}(N, \beta) = \max \left\{ I : \sum_{i=1}^I P(N, i) \leq \beta \right\} . \#$$

The concepts of resilience and CD-spectrum are closely related. From the CD-spectrum, we can get the network resilience for any β .

Consider, for example, CD-spectrum shown in Table 10.2, and let $\beta = 0.01, 0.05, 0.1, 0.3, 0.5$. Then we get:

$$\begin{aligned} \text{res}_{\text{pr}2}(N, 0.01) &= 2, \text{res}_{\text{pr}}(N, 0.05) = 6, \\ \text{res}_{\text{pr}}(N, 0.1) &= 7, \text{res}_{\text{pr}}(N, 0.3) = 9, \\ \text{res}_{\text{pr}}(N, 0.5) &= 11. \end{aligned}$$

Consider now CD-spectrum shown in Table 10.3. We get:

$$\begin{aligned} \text{res}_{\text{pr}2}(N, 0.01) &= 17, \text{res}_{\text{pr}}(N, 0.05) = 20, \\ \text{res}_{\text{pr}}(N, 0.1) &= 22, \text{res}_{\text{pr}}(N, 0.3) = 25, \\ \text{res}_{\text{pr}}(N, 0.5) &= 27. \end{aligned}$$

Note that resilience is also an invariant of the network, since it depends solely on the network topology and criterion UP/DOWN.

10.4 Birnbaum Importance Measure (BIM) and BIM-Spectrum

In this section, we introduce the Birnbaum Importance Measure (BIM) [3, 7] of network element j , $j = 1, 2, \dots, k$. Let network reliability $R(p_1, p_2, \dots, p_k)$ be a function of element reliability p_i . Then BIM of element j is defined as

$$\begin{aligned} \text{BIM}_j &= \frac{\partial R(p_1, p_2, \dots, p_n)}{\partial p_j} = R(p_1, p_2, \dots, 1_j, \dots, p_n) \\ &\quad - R(p_1, p_2, \dots, 0_j, \dots, p_n) \end{aligned} \tag{10.5}$$

BIM has a transparent probabilistic meaning: it is the gain in network reliability received from replacing a down element j by an absolutely reliable one. BIM_j gives the approximation to the network reliability δR resulted from element j reliability increment by δp_j .

Table 10.5 BIM-spectrum for network in Fig. 10.1

i	z_{i2}	z_{i3}	z_{i4}	z_{i5}
1	0	0	0	0
2	1/3	0	1/6	1/6
3	3/4	1/2	1/2	1/2
4	1	1	1	1

The use of BIM in practice is limited since usually the reliability function $R(p_1, p_2, \dots, p_k)$ is not available in explicit form. However the BIM-spectrum that we define below allows to estimate the element BIM's without knowing the analytic form of the reliability function [3].

Definition 6 Denote by Z_{ij} the number of permutations satisfying the following two conditions:

- (1) If the first i elements in the permutation are down, then the network is DOWN.
- (2) Element j is among the first i elements of the permutation.

The collection of $z_{ij} = Z_{ij}/k!$ values, $i = 1, \dots, k; j = 1, \dots, k$, is called BIM-spectrum of the network. The set of z_{ij} values for fixed j and $i = 1, \dots, k$ is called BIM_j -spectrum, or the importance spectrum of element j .

Example 2 Let us return to the network in Fig. 10.1, and using Table 10.1 calculate one of the Z_{ij} values, say Z_{42} . The permutations that satisfy the condition of the above definition are the following: (2,4,3,5), (2,4,5,3), (4,2,3,5), (4,3,5,3). Table 10.5 presents the BIM-spectrum for our network.

The columns in this table are the BIM_j spectra.

The following theorem [3] demonstrates how BIM_j can be calculated without using the reliability function.

Theorem 3 $BIM_j, j = 1, \dots, k$, equals,

$$BIM_j = \sum_{i=1}^n \frac{n!(z_{i,j} \cdot q^{i-1} p^{n-i} - (y_i - z_{i,j})q^i p^{n-i-1})}{i!(n-i)!} \tag{10.6}$$

Note that $y_k - z_{kj} = 0$, which means that in the second term of the numerator of (10.6) one can assume that index i changes from 1 to $k - 1$.

Remark 5 BIM-spectrum depends only on the network structure and the definition of network DOWN state. That is, this is invariant with respect to the up/down probabilities of the elements.

The exact calculation of BIM-spectra is a formidable task, but we can estimate the spectra using Monte Carlo approach. An appropriate algorithm [3, 5] simultaneously estimates the CD-spectrum and the BIM-spectra for all network elements.

Sometimes in the problems of analysis and design of networks, we do not need to know the values of the importance of the elements. We want to know how the elements are ranked by importance. The following theorem [3, 5] allows us to compare elements without calculating their BIM's.

Theorem 4 *Suppose the BIM's for the network are given. Let us fix two indices α and β , $\alpha \neq \beta$, and the corresponding $Z_{i,\alpha}$ and $Z_{i,\beta}$ values. Then if for all i , $i = 1, \dots, k$, $Z_{i,\alpha} \geq Z_{i,\beta}$, then $\text{BIM}_\alpha \geq \text{BIM}_\beta$ for all p values.*

For the network in Fig. 10.1, comparing the columns in Table 10.5 we get:

$$\text{BIM}_2 > \text{BIM}_4 = \text{BIM}_5 > \text{BIM}_3$$

Additional information on BIM's can be found in [8].

10.5 Border States

10.5.1 Border States and Reliability Gradient

In this section, we introduce the so-called network *border states* that are closely related to the reliability gradient.

Definition 7 Reliability gradient vector ∇R is defined as,

$$\nabla R = \left[\frac{\partial R}{\partial p_1}, \dots, \frac{\partial R}{\partial p_k} \right] \quad (10.7)$$

In words: component i of ∇R is BIM_i .

For the following definition, it is more convenient for us to determine the state of the network as a vector of element indicator variables, i.e. state $\mathbf{x} = (x_1, \dots, x_k)$, where $x_i = 1$, if element i is up, and $x_i = 0$ otherwise.

Definition 8 Network state $\mathbf{x} = (x_1, \dots, x_k) \in \text{DOWN}$ is called the neighbour of the state $\mathbf{y} = (y_1, \dots, y_k)$ if \mathbf{x} differs from \mathbf{y} in exactly one position. If $\mathbf{y} \in \text{UP}$ then we call the \mathbf{x} *border state*. The set of all border states is called the *border set* and denoted as DN^* .

Remark 6 It is clear from the definition 8 that the border state and also the border set are network invariants.

Example 3 Consider the network in Fig. 10.1. Its state is determined by the vector of nodes indicators (x_2, x_3, x_4, x_5) . (Recall that nodes 1 and 6 are terminals.) For example $(x_2 = 0, x_3 = 0, x_4 = 1, x_5 = 0) \in \text{DOWN}$ is the neighbour of $(x_2 = 1, x_3 = 0, x_4 = 1, x_5 = 0) \in \text{UP}$ (and also the neighbour of

$(x_2 = 0, x_3 = 0, x_4 = 1, x_5 = 1) \in \text{UP}$). So x is a border state. The border set for our network is

$$\text{DN}^* = \{v_1 = (0, 0, 0, 0), v_2 = (0, 1, 0, 0), v_3 = (0, 0, 1, 0), \\ v_4 = (0, 0, 0, 1), v_5 = (0, 1, 1, 0), v_6 = (0, 1, 0, 1).\}$$

To clarify the connection between border states and gradient, we introduce an *artificial evolution process* [3, 9] on network elements.

Assume that at $t = 0$ each element is down. Element i is born after random time $\tau_i \sim \exp(\mu_i)$. After the ‘birth’, the element remains up ‘forever’. Note that for fixed time t_0 , $P(\tau_i > t_0) = q_i = e^{-\mu_i t_0}$.

The following theorem [3, 9] opens the way to calculating the reliability gradient.

Theorem 5 *Let $P(v; t)$ be the probability that the network is in state v at time t . Denote by $\Gamma(v)$ the sum of μ_i over all set of indices i such that $v + (0, \dots, 1_i, 0, \dots, 0) \in \text{UP}$. Formally*

$$\Gamma(v) = \sum_{v \in \text{DN}^*, v+(0, \dots, 1_i, 0, \dots, 0) \in \text{UP}} \mu_i \tag{10.8}$$

Then the following equation holds:

$$\nabla R \bullet \{q_1 \mu_1, \dots, q_k \mu_k\} = \sum_{v \in \text{DN}^*} P(v; t) \Gamma(v), \tag{10.9}$$

where by \bullet denoted scalar product.

We see from the last equation, that knowing the probabilities of border states, we can calculate the reliability gradient. In most cases, the explicit expression of these probabilities is not available. However, formula 10.9 makes possible using the well-known Lomonosov’s algorithm [3, 9] for estimating $P(v; t)$ and ∇R .

Here we restrict ourselves to a brief description of the idea of the Monte Carlo algorithm of estimation the gradient.

First of all, we introduce an evolutionary process on network elements, as described above. We recall that for fixed time t_0 element i is in the state up with probability p_i . Now consider a sequence in an evolution process. This sequence starts from a zero state w_0 . Let $\pi = (i_1, i_2, \dots, i_k)$ be some permutation of the network elements, so that i_1 has a minimum birth time, i_2 was born the second, and so on. We associate with this permutation a sequence of network states: a state w_1 in which only i_1 is up, a state w_2 with two elements in up, and so on, up to the first state UP. This sequence of states we call the trajectory $w = (w_0, w_1, \dots, w_s)$. Consider, for example, the network with unreliable nodes is shown in Fig. 10.1. Suppose node 4 is born first, node 5 is born next and node 3 is born third. Then we get the following trajectory:

$$w = \{w_0 = (x_2 = 0, x_3 = 0, x_4 = 0, x_5 = 0),$$

$$w_1 = (x_2 = 0, x_3 = 0, x_4 = 1, x_5 = 0),$$

$$w_2 = (x_2 = 0, x_3 = 0, x_4 = 1, x_5 = 1),$$

$$w_3 = (x_2 = 0, x_3 = 1, x_4 = 1, x_5 = 1) \in UP\}$$

Note that w_2 is one of the border states.

The following is a simplified algorithm for estimating the gradient.

Algorithm 3: Evaluation of Gradient

1. **Put** $\frac{\partial R}{\partial p_1} = 0, i = 1, \dots, n$.
2. **Generate** trajectory $w = (w_0, w_1, \dots, w_s)$.
3. **Find** the first j so that w_j is a border state.
4. **Calculate convolution** $\text{Conv} = P(\tau(w_j) \leq t_0) - P(\tau(w_{j+1}) \leq t_0)$,

where $\tau(w_j)$ and $\tau(w_{j+1})$ are the birth time of w_j and w_{j+1} , respectively.

- For each** $x_i \in \Gamma(w_j)$ **calculate** $\frac{\partial R}{\partial p_1} = \frac{\partial R}{\partial p_1} + \text{Conv}$.
5. **Put** $j = j + 1$. **If** $j < s$ **Goto** 4.
 6. **Repeat** 2–5 M times.
 7. **For each** $i = 1, \dots, n$ **put** $\frac{\partial R}{\partial p_i} = \frac{\partial R}{\partial p_i} / M \cdot q_i$.

Detailed explanation of the algorithm as well as an analytical expression for calculating the convolution of exponentials are given in [3].

10.5.2 Border States and Availability

Let us now consider the following dynamic model. Each network element, independently of others, alternates between two states: up and down. When element i is up, it has failure rate λ_i . if it is down, it has repair rate μ_i . In equilibrium element i is up with probability $p_i = \mu_i / (\mu_i + \lambda_i)$. Let T_U and T_D be the average UP and DOWN periods of the network in equilibrium. Our goal is to find these periods.

It is known [4] that the network availability $Av(N)$ can be expressed as follows:

$$Av(N) = R(p_1, p_2, \dots, p_k) = \frac{T_U}{T_U + T_D} \quad (10.10)$$

The value $\rho = \frac{1}{T_U + T_D}$ is called network DOWN \rightarrow UP *transition rate*. The following theorem shows the relationship between the transition rate and the border states.

Theorem 6 It can be shown in [3, 9],

$$\rho = \sum_{v \in DN^*} P(v)\Gamma(v) \tag{10.11}$$

Example 4 Consider the network in Fig. 10.1. Assume that node i has failure rate λ_i and repair rate μ_i . Rewrite the network border set obtained above.

$$DN^* = \{v_1 = (0, 0, 0, 0), v_2 = (0, 1, 0, 0), v_3 = (0, 0, 1, 0), \\ v_4 = (0, 0, 0, 1), v_5 = (0, 1, 1, 0), v_6 = (0, 1, 0, 1)\}.$$

Now by (10.11) we get:

$$\rho = P(v_1)\mu_2 + P(v_2)\mu_2 + P(v_3)(\mu_2 + \mu_5) \\ + P(v_4)(\mu_2 + \mu_4) + P(v_5)(\mu_2 + \mu_5) + P(v_6)(\mu_2 + \mu_4).$$

For convenience, let $\mu_i = \mu, \lambda_i = \lambda$, for all $i = 2, 3, 4, 5$. Suppose that $\mu = 4, \lambda = 1$. Then $p = 0.8, q = 0.2$. Using a little arithmetics, we get $\rho = \mu(q^4 + 5pq^3 + 4p^2q^2) = 0.544$. Now, easy to get $R(N) = p + p^2 - p^3 = 0.928$. Finally using (10.10) and (10.11), we obtain: $T_U = 1.706, T_D = 0.132$.

In the case of a large network, the Lomonosov’s algorithm adapted for this purpose is used.

Let Ω be the set of all trajectories. We can rewrite (10.11) in the following form:

$$\rho = \sum_{w \in \Omega} Pr(w)P(v|w)\Gamma(v),$$

where $Pr(w)$ is the probability of the trajectory w (see [3], Chap. 9), and v is the border state determined by the trajectory w . Now, simulating the trajectories and using the corresponding variant of the Lomonosov’s algorithm we obtain the availability estimate.

Remark 7 An extremely efficient Lomonosov’s algorithm is based on ingenious graph-theoretic construction known as an evolution process on so-called Lomonosov’s ‘turnip’ [3], Chap. 9. This algorithm has a number of useful properties. Let us mention some of them.

1. The algorithm is a highly effective tool for calculating the reliability of monotone systems, for any criteria UP, and for arbitrary (not necessarily equal) element probabilities up.
2. The algorithm avoids the occurrence of a rare event phenomenon. Indeed, a distinctive feature of a rare event is that the relative error in estimating the probability of this event tends to infinity. In the Lomonosov’s algorithm, the random

choice of the trajectories does not depend on the probabilities of the elements and this explains the absence of this phenomenon.

3. It can be used to evaluate the mean stationary UP and DOWN periods.
4. It can be used to evaluate reliability gradient ∇R .

Detailed description of the algorithm and its applications can be found in the book [3], Chap. 9.

10.6 Examples

In this section, we present several examples of using the network invariants described above.

10.6.1 Network Reliability Improvement

Consider the network with unreliable nodes in Fig. 10.2. Assume that all nodes are in state up with probability $p = 0.7$. The network reliability (see Table 10.4) equals then $R = 0.6786$. Our goal is to increase network reliability to $R^* = 0.8$. Suppose it is possible to replace several nodes with more reliable ones, say with up probability $p^* = 0.9$, and we are interested in doing minimal number of such replacements. A good heuristic approach to solve this problem is the following.

First, rank all the nodes in descending order of their BIM's. Next, successively replace the nodes with more reliable ones until we get the required reliability.

The calculations performed show that all the nodes can be divided into several groups according to their importance. In particular, the first group consists of one node—29, the second group consists of four nodes: 10, 14, 24 and 28. We write it for clarity as follows:

$$\text{BIM}_{29} > (\text{BIM}_{10} = \text{BIM}_{14} = \text{BIM}_{24} = \text{BIM}_{28})$$

This conclusion is based on the analysis of the network BIM-spectra.

Further, replacing the nodes 29, 10, 14, 24 with more reliable ones, we achieve the desired reliability $R^* = 0.8169$.

Partially, the BIM-spectrum data for the nodes 1, 10, 14, 24, 28, 29 are presented in Table 10.6.

Spectrum values in the range of 20–32 are not shown. These values are almost the same, since the probability of network failure starting from step 20 is very close to 1, and at step 23 is already equal to 1. From the table, we see that the BIM spectrum values of node 29 are greater than those of the other nodes. Spectrum values for nodes 10, 14, 24, 28 are close and intertwined. Node 1 does not belong to the first two groups.

Table 10.6 Grid Network BIM-spectrum. Nodes unreliable. Terminals $T = (4,13,27,30)$

i	1	3	5	7	10	13	15	17	19
Z_1	0	0	.0010	.0101	.0810	.2144	.3990	.5089	.5923
Z_{10}	0	.0002	.0046	.0230	.1185	.3102	.4280	.5187	.5953
Z_{14}	0	.0002	.0040	.0201	.1217	.3155	.4378	.5266	.5968
Z_{24}	0	.0008	.0069	.0264	.1217	.2965	.4150	.5131	.5899
Z_{28}	0	0	.0020	.0156	.1136	.3103	.4235	.5167	.5928
Z_{29}	0	.0008	.0073	.0274	.1291	.3225	.4394	.5270	.5945

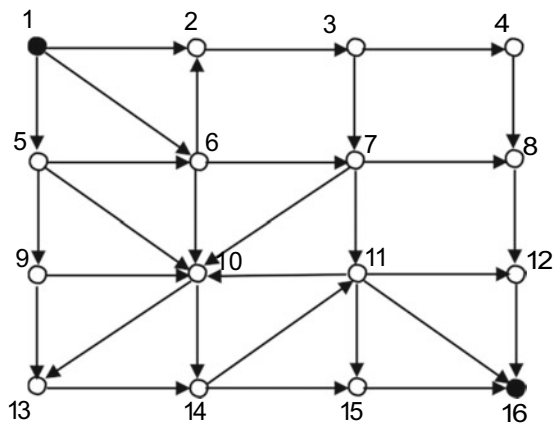
Remark 8 It should be noted that the problem described above can also be solved taking into account the cost of replacing elements. More information on network analysis and optimal network design can be found in [3, 6].

10.6.2 Resilience of Flow Network

In this section, we consider the flow network. These networks are important in many applications. By the definition, flow network is a directed network, where each edge (a, b) has a flow capacity $c(a, b)$. The flow delivered from a to b cannot exceed $c(a, b)$. Denote by s and t the source and sink nodes of the network. Denote by Maxflow the maximal flow from s to t when all edges are up. We say that the network is in DOWN state if its maximal flow is below some fixed level Φ . (Note that there exists an extensive literature with several fast algorithms for finding the maximum flow in networks.)

Let us consider now the network shown in Fig. 10.3. It has 16 reliable nodes and 30 unreliable and directed edges. The nodes 1 and 16 are the source and sink,

Fig. 10.3 Flow network with 16 reliable nodes and 30 unreliable edges



respectively. The corresponding capacities are given in Table 10.7. The calculated value of Maxflow equals 26.

Table 10.8 presents 14 values of the network spectra for $\Phi = 10$ and $\Phi = 15$. Based on these spectra, we can obtain the network resilience for different values of Φ . $\Phi = 10$ and $\Phi = 15$. Table 10.9 shows resilience for $\Phi = 10$ and $\Phi = 15$, for some values of α .

Remark 9 More detailed information on resilience of flow networks can be found in [10]. Note also that in [5] an example of comparing the resilience of networks with the same number of nodes and edges but with different topological structures is given.

Table 10.7 Edge capacities

(i, j)	$c(i, j)$	(i, j)	$c(i, j)$	(i, j)	$c(i, j)$
(1, 2)	10	(6, 2)	9	(10, 14)	7
(1, 5)	9	(6, 7)	7	(11, 10)	7
(1, 6)	10	(6, 10)	8	(11, 12)	8
(2, 3)	7	(7, 8)	6	(11, 15)	7
(3, 4)	6	(7, 10)	8	(11, 16)	11
(3, 7)	8	(7, 11)	9	(12, 16)	9
(4, 8)	7	(8, 12)	8	(13, 14)	8
(5, 6)	8	(9, 10)	8	(14, 11)	7
(5, 9)	9	(9, 13)	6	(14, 15)	7
(5, 10)	8	(10, 13)	8	(15, 16)	10

Table 10.8 Flow network CD-spectrum for $\Phi = 10$ and $\Phi = 15$

i	$y_i (\Phi = 10)$	$y_i (\Phi = 15)$	i	$y_i (\Phi = 10)$	$y_i (\Phi = 15)$
1	0	0	8	.5463	.8656
2	.0073	.0550	9	.6739	.9293
3	.0357	.1659	10	.7786	.9678
4	.0923	.3130	11	.8594	.9869
5	.1749	.4865	12	.9147	.9958
6	.2843	.6484	13	.9544	.9989
7	.4130	.7708	14	.9753	.9997

Table 10.9 Comparing resilience of flow network for $\Phi = 10$ and $\Phi = 15$

α	0.05	0.1	0.2	0.3	0.4	0.5	0.6
$\Phi = 10$	3	4	5	6	6	7	8
$\Phi = 15$	1	2	3	3	4	5	5

10.7 Concluding Remarks

Analysing the text above and the example section, we see that network sustainability analysis involves two types of information. Type A information is of non-stochastic nature and is based on network graph description, node, edge, terminal definitions and UP/DOWN definition of network states.

Four structural invariants have been defined in this paper (Signatures or Internal Distributions), CD-spectrum, BIM -spectrum and Border States) representing type A information.

All further analysis of network performance is done by combining structural invariants with information on the stochastic behaviour of network components subject to failure (edges or nodes), in static or dynamic situations. This information we call of type B. A typical example of combining A and B types of information is given in Sect. 10.6.1 on network reliability improvement.

A special ‘artificial’ variant of B-type information was an assumption that network components subject to failure fail in random and equiprobable manner imitating an external ‘shock’ situation. This shock model allows defining network resilience parameter and compares networks resilience for various versions of their structure.

In conclusion, let us note that this chapter is based on ‘binary’ approach to network structure. The book [11] goes further and introduces networks with several DOWN states. This leads to multi-dimensional invariants. Moreover, also the binary nature of failing edges or nodes can also be generalised, see [11] where in addition to up and down states of failed components, an intermediate third ‘mid’ state has been added.

References

1. Misra, Krishna B. (Ed.). (2008). *Handbook of performability engineering*. London: Springer.
2. Samaniego, F. (2007). *System signatures and their use in engineering reliability*. New York: Springer.
3. Gertsbakh, I., & Shpungin, Y. (2009). *Models of network reliability: analysis*. Combinatorics and Monte Carlo: CRC Press.
4. Gertsbakh, I. (2000). *Reliability theory with applications to preventive maintenance*. Springer-Verlag.
5. Gertsbakh, I., & Shpungin, Y. (2019). *Network reliability: a lecture course*. Springer Briefs in Electrical and Computer Engineering, Springer.
6. Gertsbakh, I., & Shpungin, Y. (2011). *Network reliability and resilience*. Springer Briefs in Electrical and Computer Engineering, Springer.
7. Birnbaum, Z. W. (1969). On the importance of different components in multicomponent system. In P. R. Krishnaiah (Ed.), *Multivariate analysis* (pp. 581–592). New York: Academic Press.
8. Gertsbakh, I., & Shpungin, Y. (2012). Combinatorial approach to computing importance indices of coherent systems. *Probability in Engineering and Information Sciences*, 26, 117–128.
9. Elperin, T., Gertsbakh, I., & Lomonosov, M. (1991). Estimation of network reliability using graph evolution models. *IEEE Transactions on Reliability*, 40(5), 572–581.
10. Gertsbakh, I., Rubinstein, R., Shpungin, Y., & Vaisman, R. (2014). Permutation methods for performance analysis of flow networks. *Probability in the Engineering and Information Sciences*, 28(1), 21–38.

11. Gertsbakh, I., Shpungin, Y., & Vaisman, R. (2014). Ternary networks. Springer Briefs in Electrical and Computer Engineering, Springer.

Ilya Gertsbakh was Professor Emeritus in the Department of Mathematics at Ben-Gurion University of the Negev, Israel. He is the author or co-author of eight books, the recent of which is 'Network Reliability: A Lecture Course' with Y. Shpungin as co-author and is published by Springer in 2019. Ilya Gertsbakh has published some 100 papers on topics in operations research, reliability, applied probability and applied statistics. He received M.Sc. degrees in Mechanical Engineering and Mathematics (1961) from Latvian State University in Riga, Latvia, and Ph.D. degree (1964) in Applied Probability and Statistics from Latvian Academy of Sciences, also in Riga.

Yoseph Shpungin is Professor Emeritus in the Software Engineering Department, at Shamoon College of Engineering, Beer Sheva, Israel. He is the co-author of four books and numerous publications in international scientific journals. He received his M.Sc. degree in Mathematics (1969) from Latvian State University in Riga, Latvia, and his Ph.D. degree (1997) in Mathematics from Ben Gurion University, Beer Sheva, Israel.