D-spectra for Networks with Binary and Ternary Components

Ilya B. Gertsbakh
Department of Mathematics,
Ben-Gurion University P. O. Box 653,
Beer-Sheva, 84105, Israel
elyager@bezeqint.net

Yoseph Shpungin
Software Engineering Department,
Sami Shamoon College of Engineering,
Beer Sheva 84100 Israel
yosefs@sce.ac.il

Radislav Vaisman
School of Mathematics and Physics,
The University of Quensland,
Brisbane 4072, Australia
r.vaisman@uq.edu.au

***Abstract*—Network reliability is an important problem from both theoretical and practical points of view. One of the most powerful approaches for handling this problem is by using a so-called D-spectra method. We describe the D-spectra for two-state networks with binary and ternary components and give an overview of its principal combinatorial properties. In particular, we establish an important connection between the D-spectra and the number of network's failure sets of a given size.**

***Keywords*—Binary and ternary components; combinatorial invariants**

I. Binary systems with binary components and their D-spectra

We will consider monotone binary systems which have two states, *UP* and *DOWN*. Their components which are subject to failure also have two states, *up* and *down*. Practically, without loss of generality, we can assume that our systems are networks. In further, our principal network model will be a triple $\mathbf{N} = (V, E, T)$, where V is the *vertex* or *node* set, $|V| = m$, E is the *edge* or *link* set, $|E| = n$, T is a set of special nodes called terminals, $|T| = h$, $T \subseteq V$. Components of the network subject to failures are the links or the nodes, except the terminal nodes. Link failure (*down*) means that the link is erased. Node failure means that all edges adjacent to it are erased, and a failed node is therefore isolated.

We say that the network is T-connected if there is a path of non failed links between each pair of terminals. Typically, we will define network *UP* state if it is T-connected and $DOWN$, otherwise.

Before we proceed, we need to introduce some notation. System (network) state we denote by a n-component binary vector

$\mathbf{x} = (x_1, x_2, ..., x_n)$, where $x_i = 1$ or 0 if component number i is up or $down$, respectively. System state is defined by a binary function $\varphi(\mathbf{x})$ which equals 1 or 0 if the system is *UP* or *DOWN*, respectively. If $\varphi(\mathbf{x}^*) = 0$, then we call $\mathbf{x}^*$ failure *vector* or *failure set*.

Now we will define the D-spectrum or signature for our system (network). Let us consider a random permutation π of component numbers

$$\pi = (i_1, i_2, i_3, ..., i_n). \quad (1)$$

Suppose that all these components are *up* and we move along the permutation from left to right and turn each component from *up* to *down*. Suppose also that network state is controlled after each step of this destruction process.

Definition 1. The ordinal number in the permutation π of the component whose turning *down* causes network to change its state from *UP* to *DOWN* is called the *anchor* of the permutation.#

Assume that the permutation π is taken randomly and equiprobable from the set of all $n!$ permutations. Then the anchor becomes a discrete random variable with the support $[1, 2, ..., n]$.

Definition 2. The distribution $f = (f_1, f_2, ..., f_n$ of the anchor is called *D-spectrum* or *signature.* "D" stands for destruction process of anchor discovery. #

Remark 1. Historically, signature was first introduced by Samaniego in [1] in a form equivalent do Definition 2 . Independently, it was introduced six years later in [2] under the term ID (Internal Distribution). The authors of [3], [4] used the term D-spectra. The book [3] introduced multidimensional signatures for networks with many states.#

Definition 3. The *cumulative* distribution function $F(k) = \sum_{i=1}^{k} f_i$ of the anchor is called the cumulative D-spectrum#.

For systems having more than , say $n = 10$ components, calculation of the D-spectra is made via its unbiased estimation obtained by using Monte Carlo.This works as follows.

1) Generate M random permutations.
2) For each permutation, carry out the "destruction" process and locate the anchor
3) Count the number of permutations M_r for having the anchor equal r, $r = 1, 2, ..., n$.
4) Take $M_r/M = \widehat{f}_r$ as an estimate for f_r and $\sum_{r=1}^{k} \widehat{f}_r$ as an estimate for $F(k)$. #

Essential acceleration of this process can be obtained using the binary search for the anchor position.

A. Principal combinatorial property of $F(k)$

Denote by $C(k)$, $k = 1, ..., n$, the number of failure sets (failure vectors) having k components *down* and the remaining $n - k$ components in state up. $C(k)$ is an important combinatorial invariant of the system because it allows to compute

978-1-4673-9941-8/16 $31.00 © 2016 IEEE
DOI 10.1109/SMRLO.2016.44

system $DOWN$ probability if we know the probabilities p and $q = 1 - p$ that a component is up or $down$, respectively. Indeed,

$$P(DOWN) = \sum_{k=1}^{n} C(k) q^k p^{(n-k)}. \quad (2)$$

An important fact is that knowing the cumulative D-spectrum it is immediately possible to obtain the number of failure sets.

Theorem 1.

$$C(k) = F(k) \frac{n!}{k!(n-k)!}.\# \quad (3)$$

There are several ways to prove this theorem. The first is purely combinatorial, see [3]. Historically, the first one uses the assumption that the system components have i.i.d. lifetimes with c.d.f. G(t). This proof can be found in [1], [5].

Note also that if we replace in (2) p by $p(t)$, the probability that system component is *up* at the time instant t, (1) becomes the probability that the system is *DOWN* at time t and therefore the probability that system lifetime τ_{sys} does not exceed t equals

$$P(\tau_{sys} \leq t) = \sum_{k=1}^{n} C(k) [q(t)]^k [p(t)]^{(n-k)}.$$

Remark 2. There is a way to describe network destruction in time without any assumption regarding component lifetime. This can be done by introducing an external *shock process* which consists of a discrete system of events (shocks) developing in time, for example, according to an arbitrary renewal process. Each shock destroys one and only one system component chosen randomly among from the set of components which are alive at the moment of shock appearance. It can be shown (see for example [3], page 24) that system lifetime θ has the following distribution function:

$$P(\theta \leq t) = \sum_{x=1}^{\infty} \rho_x F(x),$$

where ρ_x is the probability to have exactly x shocks on the interval $[0, t]$. (We set $F(x) = 1$ for $x > n$. #

II. Binary systems with ternary components and their D-spectra

We consider a system containing n components numbered from 1 to n. Each component i can be in *three* states: *up, mid, down.* These states will be denoted by numbers: 2 for *up*, 1 for *mid* and 0 for *down*. The state of system's components is described by a ternary vector

$$\mathbf{v} = (v_1, v_2, ..., v_n),$$

where $v_i = 2, 1$ or 0, according to the component state. For example, $\mathbf{v} = (2, 0, 1, 1, 2)$ means that components 1 and 5 are in state *up*, components 3,4 - in *mid,* and component 2 is *down*.

The system has only two states, i.e. is binary: *UP* and *DOWN* (denoted by 1/0, respectively). The system state is determined by a binary structure function

$$\varphi = \varphi(\mathbf{v}).$$

Our principal network model for ternary components will be a four-tuple $\mathbf{N} = (V, E, T_1, T2)$, where V is the *vertex* or *node* set, $|V| = m$, E is the *edge* or *link* set, $|E| = n$, T_1 and T_2 are two nonintersecting sets of special nodes called terminals, , $T_1 + T_2 \subseteq V$. Components of the network subject to failures are the links or nodes. We say that two nodes are *strongly* connected if there is a path of *up* links connecting these nodes. Similarly, we say that two nodes are *weekly* connected if there is a path between these nodes consisting of *up* or *mid* links. Se say that the network is *UP* if the nodes of T_1 are strongly connected *and* the nodes of T_2 are weakly connected.

We make the following standard assumptions regarding the dependence of system state on component states:

(i) $\varphi(2, 2, ..., 2) = 1$; $\varphi(0, 0, 0, ..., 0) = 0$.

(ii) if $\mathbf{v} > \mathbf{y}$, then $\varphi(\mathbf{v}) \geq \varphi(\mathbf{y})$. ($\mathbf{v} > \mathbf{y}$ means that $v_i \geq y_i$ for all i but there is at least one j such that $v_j > y_j$). **(ii)** means that the system is monotone.

Definition 4: *random permutation of* r-th *type*, $r = 0, 1, ..., n-1$.

A random permutation of component numbers $\{1, 2, ..., n\}$ in which all components are in state *mid* is called a random permutation of zero-type.

A random permutation π_r is called a permutation of r-th type, $r > 0$, if the components $i_1, ..., i_r$ on the first r positions are in state *up*, and the components $i_{r+1}, ..., i_n$ on the remaining $(n - r)$ positions are in state *mid*.

Probability of obtaining a particular ordering of component numbers $(i_1, i_2, ..., i_n)$ in the r-th type permutation is $1/n!$#

Definition 5: *failure (cut) set* and $(r; x)$*-failure set*.

Failure set is a vector $\mathbf{v} = (v_1, v_2, ..., v_n)$ of component states such that

$$\varphi(\mathbf{v}) = \varphi(v_1, v_2, ..., v_n) = 0.$$

Here $v_i = 2, 1$ or 0 means that vector $\mathbf{v}$ component i is in state up, mid or $down$, respectively.

A failure set which has r components in up, x components in *down*, and $(n - r - x)$ components in *mid* is termed a $(r; x)$*-failure set*.#

A. Destruction process

The destruction process has several stages denoted $0, 1, ..., n-1$. Stage r of destruction process consists of:

(a) generating a random permutation of r-th type;

(b) an initial check of system state, and

(c) sequential destruction of its *mid* components (i.e. turning them from *mid* to *down*) by moving from left to right.

In (a), according to Definition 1, we generate a random permutation of components numbers, assign to the first r of them state *up* and to the remaining - state *mid*. For permutation of zero-type all components are in state *mid*.

(b) means checking system state when the r first components in the permutation are *up* and *all* remaining components are in *mid*. If for a particular r-permutation π_r the check reveals that the system is already *DOWN*, we say that this permutation has *anchor* equal zero.

Stage (c) is carried out only if the anchor is *not* zero. It consists of turning the components from *mid* to *down* by moving from left to right, and checking the system state after each component destruction.

Definition 6: *anchor of an r-type permutation*.

The anchor of a permutation π_r of r-th type denoted $\delta(\pi_r)$ is the number of components which have been turned *down* when the system was for the first time discovered in state *DOWN*.#

If in an r-type permutation, the system is already *DOWN* when no component has been turned from *mid* to *down*, then this permutation has anchor equal zero.

Remark 3. There might exist such r for which the permutation has no anchor. It means that after destruction of all $n-r$ *mid* components the system remains in *UP*. If this happens for a particular $r^\star$, then by monotonicity property of the system, there will be no anchor for all $r > r^\star$.

It is important to note that by the same monotone property of the system, for any permutation there might be at most a single transition $UP \to DOWN$, i.e. at most a single anchor. This property is very important for designing an fast search procedure for locating the anchor.#

Let $f_r(y)$ be the probability that the anchor of r-th type permutation equals y, $y = 0, 1, 2, ..., n-r$:

$$f_r(y) = P(\text{In } \ \pi_r, \text{the anchor } \delta(\pi_r) = y).$$

Since the total number of permutations for each r is $n!$, and all permutations are equally probable,

$$f_r(y) = \frac{\text{number of } r\text{-permutations with } \ \delta(\pi_r) = y}{n!}, \quad (4)$$

for $y = 0, 1, \ldots, n-r$.

Definition 7: *cumulative r-spectrum*.

$$F_r(x) = \sum_{y=0}^{x} f_r(y), x = 0, 1, 2, ..., n-r,$$

is called cumulative r-spectrum. Obviously, $F_r(x) \leq 1$.#

Remark 4. Let us assume that all r-type permutations have anchor equal zero. It means that the numerator of $f_r(0)$ in (2) will be equal $n!$ and therefore $f_r(0) = 1, f_r(j) = 0$ for $j = 1, ..., n-r$ and $F_r(0) = 1 = F_r(j), j = 1, ..., n-r$.

Definition 8: *ternary D-spectrum*.

The collection of all cumulative r-spectra $\mathcal{T}sp = \{F_r(x)\}$ for $0 \leq r < n$ is called *ternary D-spectrum*.#

B. Principal combinatorial property of ternary D-spectrum

Denote by $C(r; x)$ the number of all $(r; x)$ - failure sets in the system. (Let us remind that an $(r; x)$ -failure set has r components *up*, x components *down* and the remaining components in *mid*.)

Theorem 2

$$C(r; x) = F_r(x) \cdot \frac{n!}{r!x!(n-r-x)!}.\# \quad (5)$$

For combinatorial proof of this theorem see [4].

Theorem 1 opens way to compute system *DOWN* probability for the case that all components are independent and have identical probabilities (p_2, p_1, p_0) for *up, mid* and *down* states, respectively. In that case each $(r; x)$ - failure set has probability $p_2^r p_1^{(n-r-x)} p_0^x$ and therefore the probability weight of all such sets equals

$$C(r; x) \cdot p_2^r p_1^{(n-r-x)} p_0^x. \quad (6)$$

Theorem 3

$$P(DOWN) = \quad (7)$$
$$= \sum_{\{x \geq 0, r \geq 0 : 0 \leq r+x \leq n\}} C(r; x) \cdot p_2^r p_1^{(n-r-x)} p_0^x.\#$$

This formula can be rewritten in an equivalent "dynamic" form to include the time factor. Suppose we have n independent and identically distributed stochastic processes $\{\chi_i(t), t > 0, i = 1, ..., n\}$. Each $\chi_i(t)$ is a decreasing, left continuous process with three states: 2, 1 and 0. State 0 is absorbing. Each trajectory of $\chi_i(t)$ starts at $t = 0$ in state 2, jumps into state 1 and later gets absorbed in state 0. At any time instant $t > 0$, $\chi_i(t)$ is in one of its three states with probabilities $p_2(t), p_1(t)$ and $p_0(t)$, respectively. Obviously

$$p_0(t) + p_1(t) + p_2(t) = 1, t > 0.$$

Let $\tau_2(i)$ be the sojourn time of $\chi_i(t)$ in state 2. Then $P(\tau_2(i) \leq t) = 1 - p_2(t)$. Let $\tau_1(i)$ be the sojourn time of $\chi_i(t)$ in state 1. Then the event $\{\tau_2(i) + \tau_1(i) \leq t\}$ means that at time $t + 0$, $\chi_i(t)$ has already left state 1, and is therefore in state 0, i.e.

$$P(\tau_2(i) + \tau_1(i) \leq t) = p_0(t).$$

Note that if at time instant t the system is *DOWN*, then its failure-free operation time τ_{UP} does not exceed t. Then we can write that

$$P(\tau_{UP} \leq t) = \quad (8)$$
$$= \sum_{\{x \geq 0, r \geq 0 : 0 \leq r+x \leq n\}} C(r; x)[p_2(t)]^r [p_0(t)]^x \cdot$$
$$\cdot [1 - p_0(t) - p_2(t)]^{(n-r-x)}.\#$$

III. Concluding remarks: unsolved problems

1) When dealing with networks having *binary* components, it is quite natural to introduce networks with many states [4]. For example, the network initially is fully connected. When nodes of this network start failing, the maximal component of this network is decreasing. Initial state of the network is defined as *UP*. When maximal component becomes less or equal 0.7 of all nodes, we say that network went into state $DOWN1$. When the maximal component becomes less or equal 0.5 of all nodes, we say that the network entered state $DOWN2$. Finally, when the maximal component is less or equal 0.25 of all nodes, we say that the network is in state $DOWN0$, (in "total failure" state).
   (Here we must made an agreement that network state is determined by its lowest state. Suppose, network initially has 100 nodes, Suppose, the maximal component now has 30 nodes. So, formally the network is in state *DOWN1*,and also in *DOWN2* but not in *DOWN0*. So, to the network with 30 nodes is declared to be in state $DOWN2$.)
   Introducing network with many states for *binary* components was done via determining an ordered sequence of anchors for each random permutation of component numbers. When we start the destruction process, we fix the ordinal numbers of the components when the transitions $UP \to DOWN1 \to DOWN2 \to DOWN0$ took place.
   It is an open problem how a similar process can be defined for ternary components. The main complication arises from the "multidimensional" structure of the ternary D-spectrum.
2) For networks with binary components, we defined a shock process, which allows to introduce "shock time" for determining network lifetime. This was done without having any information about component lifetimes,their dependence or independence. All we needed to postulate the existence of an external sequence of events (shocks) and a agreement that each shock "kills" randomly one of network components which is alive at the instant of shock appearance.
   Can anything similar be done for a network with ternary components? How to distinguish which shock kills a component completely or just causes its transition from *up* to *mid*?
   We hope that that in future it will become possible to find answers to these questions.

Acknowledgment

This work of Radislav Vaisman was supported by the Australian Research Council Centre of Excellence for Mathematical & Statistical Frontiers, under grant number CE140100049.

References

[1] F. J. Samaniego. On closure under ifr formation of coherent systems. IEEE Transactions on Reliability, 34:69–72, 1985.
[2] T. Elperin, I. B. Gertsbakh, and M. Lomonosov. Estimation of network reliability using graph evolution models. IEEE Transactions on Reliability, 40(5):572–581, 1991.
[3] I. B. Gertsbakh and Y. Shpungin. Network Reliability and Resilience. SpringerBriefs in Electrical and Computer Engineering. Springer, 2011.
[4] I. B. Gertsbakh, Y. Shpungin, and R. Vaisman. D-spectrum and reliability of a binary system with ternary components. Probability in Engineering and Informational Sciences, 30:1–15, 2016.
[5] F. J. Samaniego. System Signatures and Their Applications in Engineering Reliability. International Series in Operations Research & Management Science. Springer, 2007.