Gheorghe M.Panaitescu

FIABILITATE SI DIAGNOZĂ

Note de curs

Universitatea "Petrol-Gaze" Ploiesti
Catedra Automatică si calculatoare
2007

CUVÂNT INTRODUCTIV

Lucrarea aceasta reprezintă suportul cursului de Fiabilitate si diagnoză*, tinut pe durata unui semestru, câte două ore pe săptămână, la anul V, specializarea* Automatică si informatică industrială *din cadrul Facultătii de inginerie mecanică si electrică a Universitătii "Petrol-Gaze" Ploiesti. Este produsul unei experiente de peste zece ani. Versiunea de fată este revăzută si adăugită pentru instruirea celei de a cincisprezecea promotii de la specializarea amintită. Într-un plan de învătământ mai recent, acest curs a fost introdus ca disciplină optională si la anul V, specializarea* Electronică*.*
Textul care urmează nu se constituie într-un manual *de* Fiabilitate si diagnoză*.*
După cum arată si subtitlul, este nu mai mult decât Note de curs *menite a ghida expunerile celui care predă disciplina cu acelasi nume sau, în cel mai bun caz, o referintă concisă a initiatilor în domeniu. Nu sunt continute toate explicatiile si comentariile care cu sigurantă ar fi necesare pentru ca textul să devină un manual, comentarii care se fac însă la expunerea orală. Asadar, pentru studenti, lectura fie si foarte atentă a celor ce urmează nu poate suplini audierea cursului. Dacă într-o formă sau alta această lucrare se difuzează, se difuzează numai ca un ajutor în întelegerea lecturii notitelor proprii în vederea pregătirii verificărilor partiale si a verificării finale prevăzute la această disciplină. Studentii sunt îndemnati să consulte concomitent bibliografia indicată pentru subiectele care nu se regăsesc în continuare dar si pentru subiectele care sunt preluate din referinte si reformulate în* Notele de curs *de mai jos.*
Studentii au la dispozitie si un Ghid de lucrări *la disciplina* Fiabilitate si diagnoză*. Acesta este atasat prezentelor* Note de curs *dar este afisat si pe serverele catedrelor Automatică si calculatoare si Electrotehnică si electronică în scopul accesului "on line" în timpul aplicatiilor la această disciplină.*

C U P R I N S

NOTIUNI INTRODUCTIVE

Sistemele *hardware* si *software* sunt create uzual pentru a îndeplini anumite sarcini, pentru a atinge anumite obiective de natură tehnică-tehnologică, din domeniul cunoasterii etc. Este foarte important ca aceste sisteme să functioneze adecvat, adică întreruperile nedorite, necomandate să fie cât mai rare si cât mai scurte, iar dacă se produc, depanarea sau înlocuirea să fie posibile, măcar una dintre ele si să nu fie excesiv de îndelungate. Desigur, toate aceste conditii trebuie satisfăcute nuantat deoarece totdeauna sunt implicate costuri. Nu este nici pe departe necesar a se crea sau a se achizitiona un aparat capabil să functioneze practic fără cusur ani la rând dacă utilizarea lui vizează câteva săptămâni. Un asemnea aparat ar costa foarte mult. În asemenea împrejurări, este rational a uza de unul mai ieftin, mai putin durabil, dar care în acele săptămâni este suficient de sigur pentru a servi atingerii telului propus. Problema readucerii sistemului defect la parametrii functionali normali în raport cu obiectivul urmărit se poate face, asa cum în treacăt s-a spus, prin operatii de depanare sau prin înlocuirea integrală. Si aici trebuie cumpănit prin prisma costurilor: depanarea poate costa uneori mai mult decât înlocuirea, alteori depanarea pur si simplu nu este posibilă.

Timpul necesar depanării unui sistem care subit devine nefunctional include si o prealabilă diagnosticare care ea însăsi are o durată uneori semnificativă. Un echipament sau un program de calcul defect nu trebuie demontat, reanalizat în întregime ci numai în acea parte a lui sau în acea reuniune de părti vinovată de proasta functionare sau de nefunctionare. Din nou, diagnoza corectă este o problemă care implică importante cheltuieli de bani si de timp. Readucerea la standardul functional necesar depinde în mare măsură de iscusinta cu care este pus diagnosticul. Este aproape de la sine înteles că punerea diagnosticului si remedierea defectelor nu sunt totdeauna faze succesive. Uneori faza de diagnosticare merge paralel si se împleteste cu operatiile de depanare propriu-zisă.

În legătură cu functionarea sau nefunctionarea sistemelor, fie ele *hardware* sau *software*, sunt câteva concepte care trebuie definite cel putin provizoriu încă de pe acum. Astfel, se vorbeste de **capacitatea operatională** a unui sistem în functiune, care nu este altceva decât capacitatea acelui sistem de a îndeplini anumite cerinte operationale, într-un interval de timp dat, în conditii specificate. **Fiabilitatea** în sens larg sau **disponibilitatea** unui sistem constă în capacitatea lui de a îndeplini corect functiunile pentru care este gândit, la un moment dat sau pe un interval de timp precizat, dacă sistemul este folosit, exploatat în anumite conditii si dacă este întretinut corespunzător. **Mentenabilitatea** este capacitatea sistemului de a putea fi mentinut sau repus în functiune într-un timp

precizat dacă întretinerea sau repararea sunt făcute urmând anumite proceduri recomandate si folosind resursele prescrise. **Securitatea** unui sistem este capacitatea de a prezerva starea de sănătate a oamenilor, de a nu pune în pericol valori materiale prin functionare defectuoasă.
Un sistem poate fi compus din mai multe subsisteme. Functionarea fiecărui subsistem se reflectă într-un anumit mod în functionarea ansamblului. Relatia *întreg-parte*, *sistem-componentă* nu poate fi totdeauna definită univoc. În principiu orice sistem este alcătuit din părti. Detalierea în părti este de cele mai multe ori la alegerea analistului de sistem. Frecvent părtile corespund unor subunităti structurale clar diferentiabile fizic.
Functionarea sistemului este, asa cum s-a spus, într-o anumită relatie cu functionarea părtilor dar nu neapărat defectarea unei părti coincide cu scoaterea din functie a întregului sistem. Sistemul poate functiona uneori si cu unele părti ale lui defecte. Asadar, sistemul poate avea anumite redundante constructive create de cele mai multe ori cu premeditare, care fac ca unele părti să poată suplini alte părti nefunctionale la un moment dat. Desigur, si redundantele costă dar ele pot contribui la o importantă crestere în siguranta în functionare a sistemului, de cele mai multe ori cu cheltuieli semnificativ mai mici decât cele asociate unui sistem fără redundante dar foarte rafinat.
Această enumerare sumară de aspecte legate de functionarea în sigurantă a sistemelor *hardware* sau *software* fără deosebire decât cel mult în nuante dau o imagine destul de cuprinzătoare a obiectului si obiectivelor acestui curs de **Fiabilitate si diagnoză**.
Definirea bunei funcționări si a defectărilor nu este universală. În toate cazurile functionarea si nefunctionarea sunt situatii/evenimente contrarii. În sens cuprinzător, buna functionare a unui sistem corespunde îndeplinirii unui set de obiective conform destinatiei prin proiect a respectivului sistem. Obiectivele înseşi trebuie definite precis pentru a putea defini apoi corect buna functionare a sistemului.
Defectiunile pot fi clasificate în diferite moduri. Dacă se consideră momentul aparitiei lor defectiunile pot fi:

a) ***infantile***, dacă apar în perioada de exploatare de început;
b) ***de îmbătrânire***, dacă sunt datorate uzurii componentelor sistemului;
c) ***accidentale***, dacă sunt datorate unor solicitări bruste, întâmplătoare; acestea au o frecventă mai mică decât cele din celelalte categorii.

Alte posibile clasificări ale defectiunilor sunt date în continuare:

Conditiile aparitiei	În conditii normale, în conditii anormale
Provenienta	Din proiectare, din executie, din exploatare
Posibilitatea eliminării cauzelor	Eliminabile, neeliminabile
Posibilitatea de utilizare ulterioară a sistemului	Utilizare totală, utilizare partială
Mijlocul de eliminare	Prin schimbarea elementului defect, prin reglare

Frecventa aparitiei	Unică, repetată
Posibilitatea de prognoză	Neprognozabile, prognozabile
Complexitatea interventiilor pentru eliminare	Interventii simple, interventii complexe
Consecintele	Primejdioase, majore, neprimejdioase, minore
Modul de depistare	Defectiuni vizibile, defectiuni ascunse
Gradul de dependentă între defectiuni	Defectiuni dependente, defectiuni mutual independente
Modificarea caracteristicilor functionale	Modificare bruscă, modificare lentă

Pentru mentinerea în functie a sistemelor sau înlocuirea lor oportună există desigur politici, de la caz la caz, în parte sau total diferite. Cursul prezent încearcă să stabilească câteva modele adecvate pentru procesul de deteriorare a calitătilor functionale ale sistemelor *hardware* si/sau *software* si câteva modalităti de detectare si de diagnosticare a defectiunilor posibile. Nici utilizarea redundantelor în sistemele complexe si implicit o anumită tolerantă la defecte nu este ignorată. Autorul acestor *Note de curs* nădăjduieste că, pornind de la aceste notiuni mai curând sumare de fiabilitate si diagnoză, viitorii ingineri automatisti sau electronisti vor putea dezvolta propriile lor metode si mijloace de tratare a problematicii complexe din domeniu.

COMPLEMENTE DE TEORIA PROBABILITĂTILOR SI STATISTICĂ MATEMATICĂ

Siguranta în functionare a diverselor sisteme are un vădit caracter aleator. Starea de functionare sau nefunctionare la un moment dat este imprevizibilă în sens determinist dar cuantificabilă sub aspectul stării probabile a sistemului la acel moment. De aceea, în capitolul prezent sunt (re)aduse în discutie câteva elemente de teoria probabilitătilor si de statistică matematică absolut necesare în întelegerea si tratarea consistentă a problemelor de fiabilitate si diagnoză.

Spatiul evenimentelor

Se notează cu E spatiul evenimentelor – multimea evenimentelor posibile relative la un experiment.
Exemplu: dacă experimentul constă în observarea stării de functionare a unui sistem atunci cele două rezultate posibile, *sistemul este functional* si *sistemul este disfunct* sunt evenimente.
Între evenimente poate avea loc o relatie de implicatie, scrisă $A \subset B$.
Implicatia constă în regula: producerea evenimentului A conduce la producerea necesară a evenimentului B. Implicatia reciprocă, $A \subset B$ si $B \subset A$ înseamnă egalitatea sau echivalenta celor două evenimente.
Cu evenimente se pot face operatii, două binare si una unară, care au ca rezulatate alte evenimente. Acestea sunt respectiv:
Reuniunea, notată $A \cup B$, cu rezultatul un eveniment care constă în producerea a cel putin unuia din cele două evenimente, A sau B;
Intersectia, notată $A \cap B$, cu rezultatul un eveniment care constă în producerea ambelor evenimente concomitent, si A, si B;
Luarea *complemetarului* sau a *contrarului* unui eveniment A, notat cu $\overline{A}$, care face dintr-un eveniment contrarul lui.
Operatiile binare sunt asociative si comutative si pot fi iterate pentru mai mult de două evenimente.
Între evenimentele unui spatiu se introduc si două evenimente speciale, $\emptyset$ – evenimentul imposibil si E – evenimentul sigur.
Relatia $A \cap B = \emptyset$ exprimă incompatibilitatea mutuală a celor două evenimente. Producerea unuia exclude producerea celuilalt.
E este o multime partial ordonată, relatia de ordine este implicatia. Evenimentele limită inferioară în sirurile ordonate complete se numesc *atomi* sau *evenimente elementare*. Celelalte evenimente sunt evenimente compuse. Ele

se obtin din alte evenimente, în particular din cele elementare, prin operatiile definite mai sus.
O multime de evenimente E împreună cu operatiile de reuniune, de intersectie si de luare a complementarului, cu evenimentul imposibil si evenimentul sigur incluse se organizează ca o *algebră booleană*.
Fie Ω multimea tuturor evenimentelor atomice sau elementare dintr-o multime de evenimente finită E. Evident $\Omega \neq \emptyset$. Multimea Ω si evenimentele compuse obtinute prin reunirea si intersectarea evenimentelor elementare se organizează ca un corp. O submultime a multimii de părti ale multimii atomice Ω, $K \subset P(\Omega)$ se organizează, de asemenea, ca un corp dacă

$$A \in K \Rightarrow \overline{A} \in K$$
$$A, B \in K \Rightarrow A \cup B \in K$$
$$A, B \in K \Rightarrow A \cap B \in K$$

În aceste coditii perechea (Ω, K) este un corp de evenimente si este un σ-corp sau corp borelian dacă orice reuniune sau intersectie de evenimente din K, finită sau infinită apartine multimii K.
Într-un spatiu E complet si atomic, orice eveniment $A \neq \emptyset$ se poate scrie ca o reuninune de elemente din Ω:

$$A = \bigcup_{\omega_i \in \Omega} \omega_i$$

O multime de evenimente $A_i \in K$ (i = 1, 2, …, n) mutual incompatibile, altfel spus care satisfac relatiile $A_i \cap A_j = \emptyset$ pentru $A_i, A_j \in K$ cu $i \neq j$, este o *partitie* a unui eveniment $A \in K$ dacă

$$\bigcup_{i=1}^{n} A_i = A$$

Dacă $A = \Omega$ atunci evenimentele din familia $A_i \in K$ (i = 1, 2, …, n) cu proprietătile mentionate alcătuiesc un *sistem complet* de evenimente.

Probabilităti

Pe multimea evenimentelor dintr-un corp K se defineste o functie reală P numită *probabilitate*, care are proprietătile:

$$P(A) \geq 0 \text{ pentru } \forall A \in K$$
$$P(\Omega) = 1$$
$$P\left(\bigcup A_i\right) = \sum P(A_i)$$

cu reuniunea si suma pe toate valorile indicelui i ai unei familii de evenimente $A_i \in K$, două câte două mutual incompatibile, adică $A_i \cap A_j = \emptyset$ ori de câte ori $i \neq j$.
Dacă ultima proprietate are loc si pentru reuniuni numerabile atunci probabilitatea P se numeste complet aditivă (sau σ-aditivă) pe corpul (borelian) de evenimente (Ω, K).

Tripletul (Ω, K, P) se numeste câmp (borelian) de probabilitate. Dacă Ω este o multime finită atunci (Ω, K, P) este un câmp de probabilitate discret.
Probabilitatea P are alte câteva proprietăti derivate din cele trei de mai sus.
Probabilitatea evenimentului imposibil

$$P(\emptyset) = 0$$

Relatia dintre probabilitătile evenimentelor contrare

$$P(A) = 1 - P(\overline{A})$$

Probabilitatea evenimentului diferentă, $A - B = A \cap \overline{B}$

$$P(A - B) = P(A) - P(A \cap B)$$

Limitele inferioară si superioară pentru functia P

$$0 \le P(A) \le 1$$

Probabilitatea evenimentului diferentă simetrică, $A \Delta B = (A - B) \cup (B - A)$

$$P(A \Delta B) = P(A) + P(B) - 2P(A \cap B)$$

Probabilitatea reuniunii a două evenimente oarecare

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

O extindere a relatiei ultime pentru reuniunea a n evenimente oarecare este

$$P\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{j=1}^{n} (-1)^{j+1} S_j \quad \text{cu} \quad S_j = \sum_{i_1, i_2, \ldots, i_j \le n} P(A_{i_1} \cap \ldots \cap A_{i_j}) \quad j \le n .$$

Dacă $F = \{A_i\}_{i \in I}$ este o familie numerabilă de evenimente mutual exclusive atunci $P\left(\bigcap_{i \in I} A_i\right) = 0$. Dacă familia $F = \{A_i\}_{i \in I}$ este si exhaustivă, adică este un sistem complet de evenimente, atunci $P\left(\bigcup_{i \in I} A_i\right) = 1$.

Probabilităti conditionate

Evenimentele se pot conditiona reciproc. Producerea unui eveniment poate modifica probabilitatea de producere a unui alt eveniment. Relatia de bază pentru calculul unei probabilităti conditionate este

$$P_B(A) = P(A/B) = P(A \cap B)/P(B)$$

cu evenimentul care conditionează trecut ca indice sau, în argument, după caracterul despărtitor "/".
În general,

$$p(A/B) \ne P(A) \text{ si } P(B/A) \ne P(B)$$

ceea ce indică o dependentă între cele două evenimente. Dacă are loc egalitatea în ambele cazuri, atunci evenimentele sunt independente.
Dacă probabilitatea unei intersectii finite de evenimente este nenulă

$$P\left(\bigcap_{i=1}^{n} A_i\right) \ne 0$$

atunci probabilitatea respectivă se poate calcula cu formula

$$P\left(\bigcap_{i=1}^{n} A_i\right) = P\left(A_n / \bigcap_{i=1}^{n-1} A_i\right) \ldots P(A_2 / A_1) P(A_1)$$

care se demonstrează inductiv.

Dacă $(A_i)_{i=\overline{1,n}}$ este o partitie a câmpului Ω atunci probabilitatea unui eveniment oarecare se poate calcula cu relatia

$$P(A) = \sum_{i=1}^{n} P(A_i) P(A / A_i)$$

cunoscută ca *formula probabilității totale*.

Mai este de retinut *formula lui Bayes*:

$$P(A_i / A) = P(A_i) P(A / A_i) / \sum_{i=1}^{n} P(A_i) P(A / A_i)$$

care în aceleasi conditii, $(A_i)_{i=\overline{1,n}}$ o partitie a câmpului **Ω**, permite calculul probabilității fiecărui eveniment al partitiei conditionat de evenimentul $A \in K$, altfel oarecare.

Variabile aleatoare

Variabilă aleatoare este o functie $X: \Omega \to R$ astfel încât

$$\{X < x\} \Rightarrow \{\omega \in \Omega / X(\omega) < x\} \in K \quad \forall x \in R$$

Un exemplu de variabilă aleatoare îl constituie functia indicator a unui eveniment $A \in K$

$$\chi_A = \begin{cases} 0 & \omega \notin A \\ 1 & \omega \in A \end{cases}$$

Dacă *X* este o variabilă aleatoare definită pe câmpul (**Ω**, *K*, *P*) atunci pentru oricare două valori $x_1, x_2 \in R, x_1 \le x_2$ toate intervalele finite sau infinite delimitate de cele două valori corespund unor evenimente din *K* si, prin generalizare, pentru orice multime *I*, reuniune de intervale din *R*, se poate calcula

$$P_X(I) = P[X(\omega) \in I] = P[X^{-1}(I)]$$

Functia $P_X(I)$ este distributia de probabilitate a variabilei aleatoare *X*. Se poate vorbi asadar de P_X ca de o probabilitate definită pe câmpul (R, K_X) în care $K_X = \left\{I \subset R / X^{-1}(I) \in K\right\}$.

Dacă variabila aleatoare *X* ia valori într-o multime cel mult numerabilă

$$\{x_i / x_i \in R, i \in I, I \subset N^+\}$$

atunci ea se numeste discretă si

$$\sum_{i \in I} P_X(x_i) = 1$$

$$P_X(J) = \sum_{x_i \in J} P_X(x_i) \quad \forall J \in K_X$$

Dacă *X* variază continuu pe un interval $I \in K_X$ atunci

$$P_X(I) = \int_I f_X(x)dx$$

si este o functie absolut continuă.

Functia $f_X(x)$ este *densitatea de probabilitate* sau *densitatea de repartitie* a variabilei *X*, este nenegativă pentru orice x si are proprietatea

$$\int_{-\infty}^{\infty} f_X(x)dx = 1$$

Functia care urmează se numeste *functie de repartitie* a variabilei aleatoare *X*

$$F_X(x) = P[X(\omega) < x] = P_X[(-\infty, x)]$$

Functia este nedescrescătoare pe întreaga axă reală

$$a < b \Rightarrow F_X(a) \le F_X(b) \quad \forall a,b \in R$$

si este continuă la stânga în fiecare punct

$$\lim_{x \to a, x < a} F_X(x) = F_X(a) \; \forall a \in R$$

Valorile minimă si maximă sunt date de

$$\lim_{x \to -\infty} F_X(x) = 0 \qquad \lim_{x \to \infty} F_X(x) = 1$$

Eventualele discontinuităti sunt de speta primă si sunt cel mult numerabile.

Orice functie cu proprietătile de mai sus are corespondent un câmp de probabilitate.

Pentru o variabilă aleatoare discretă functia de repartitie este o functie în scară

$$F_X(x) = \sum_{x_i < x} P_X(x_i)$$

Pentru o variabilă aleatoare continuă sunt valabile în general relatiile

$$F_X(x) = \int_{-\infty}^{x} f_X(x)dx, \; f_X(x) = \frac{d}{dx} F_X(x)$$

Pentru orice interval $[a,b) \subset R$

$$P_X\{[a,b)\} = F_X(b) - F_X(a) \quad \text{si} \quad P_X(a) = 0$$

Se notează cu $V(\Omega, K, P)$ multimea tuturor variabilelor aleatoare definite pe câmpul de probabilitate (Ω, K, P). Evident, există foarte multe asemenea variabile.

Dacă variabilele aleatoare $X, Y \in V(\Omega, K, P)$, atunci suma, produsul celor două variabile aleatoare, modulul, puterea, în general o functie măsurabilă Borel de oricare dintre ele sunt variabile aleatoare din multimea $V(\Omega, K, P)$.

Dependenta a două variabile aleatoare din aceeasi familie $V(\Omega, K, P)$ se poate exprima printr-o formulă asemănătoare cu cea a probabilitătii conditionate a evenimentelor:

$$P(X \in A / Y \in B) = P(X \in A, Y \in B) / P(Y \in B)$$

cu $A \in K_X$ si $B \in K_Y$. Într-o manieră asemănătoare se pot defini functii de repartitie conditionate.

Variabilele aleatoare sunt descrise uneori prin asa-numitele *momente*.
Momentele ordinare se calculează cu relatia

$$m_r = \sum_i p_i x_i^r$$

dacă este vorba de o variabilă aleatoare discretă si suma se calculează pe toate valorile x_i pe care variabila le poate lua, sau cu relatia

$$m_r = \int_{-\infty}^{+\infty} x^r f(x)dx$$

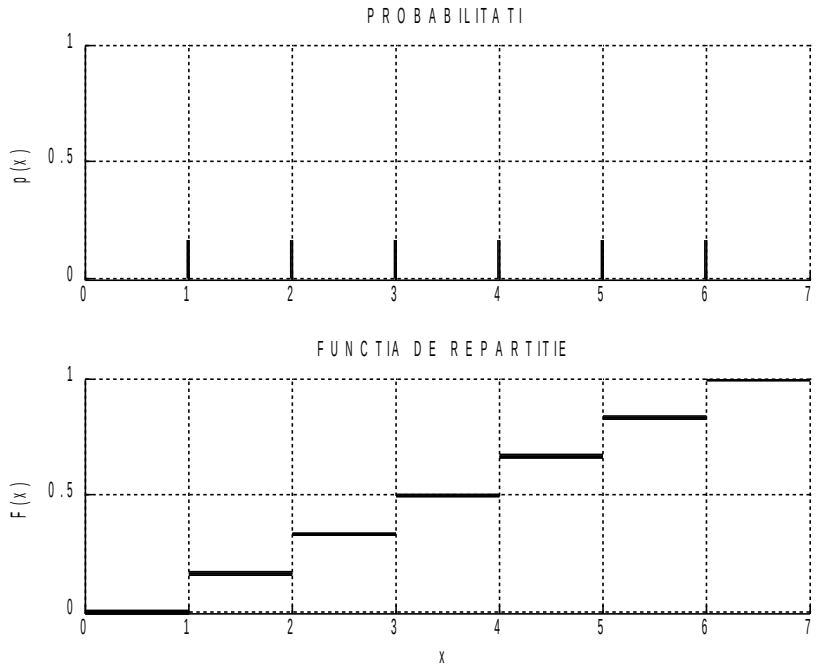
dacă variabila este de tip continuu. Numărul *r* este totdeauna un număr natural si se vorbeste de momentul de ordinul *r* al variabilei, trecut în formulele date si ca indice al momentului calculat.
Între momentele obisnuite, ordinare se retine momentul de ordinul 1, care se mai numeste si *media* variabilei aleatoare. În notatia curentă indicele $r = 1$ se omite si de aceea pentru medie se utilizează deseori notatia *m*.
Momentele centrate sunt similare celor ordinare dar sunt calculate pentru abaterile de la media *m* definită mai devreme. Astfel, pentru variabile discrete se scrie

$$M_r = \sum_i p_i (x_i - m)^r$$

pentru variabilele continue se scrie

$$M_r = \int_{-\infty}^{+\infty} (x - m)^r f(x)dx$$

iarăsi cu *r* un număr natural.
Cazul $r = 2$ particularizează momentul centrat la asa-numita dispersie sau variantă a variabilei aleatoare, care este notată uzual cu σ^2.
Media si dispersia sunt importante sub aspect practic pentru că sunt valori care dau măsura modului în care valorile unei variabile aleatoare se grupează. Valorile efectiv luate de o variabilă aleatoare se grupează în jurul mediei, iar gruparea aceasta poate fi mai strânsă sau mai largă după cum dispersia variabilei este mai mică sau mai mare. Din observatii experimentale se pot evalua media valorilor observate, o medie experimentală, care tinde în probabilitate către media teoretică *m* si o dispersie empirică estimatoare a dispersiei (teoretice) σ^2.
Sunt date în continuare câteva exemple de legi de repartitie pentru variabile aleatoare de tipuri variate.
O repartitie discretă uniformă este aceea asociată zarului perfect aruncat pe o suprafată plană orizontală. Celor sase evenimente atomice asociate cu cele sase fete ale zarului li se pot asocia valori oarecare, reale, în număr de cel mult sase, în particular chiar numărul de puncte, 1, 2, 3, 4, 5 sau 6, observate pe fata de deasupra la fiecare aruncare. Diagramele alăturate arată probabilitătile asociate valorilor pe care variabila le ia efectiv si functia ei de repartitie.



O variabilă aleatoare asociată cu zarul perfect

Alte câteva legi de repartitie teoretice, foarte utilizate si în modelarea fiabilitătii sistemelor sunt prezentate pe scurt în continuare.
Dintre legile de repartitie pentru variabile aleatoare discrete sunt de mentionat legea binomială sau legea lui Bernoulli si legea lui Poisson.
Legea binomială se referă la o variabilă aleatoare *m* care ia un număr finit de valori exclusiv în multimea numerelor naturale. Probabilitătile sunt calculate cu relatia

$$P(m) = C_n^m p^m (1 - p)^{n-m}$$

cu $0 \le m \le n$ si p un număr în intervalul [0,1]. Se observă si motivul pentru care legea se numeste "binomială": probabilitătile $P(m)$ sunt termeni din dezvoltarea puterii a n-a a binomului $[p + (1 - p)]$. Variabila aleatoare m are media np si dispersia $np(1 - p)$. Modelul fizic generator al acestor numere naturale aleatoare îl constituie o urnă cu bile de două culori. Evenimentele constau în extragerea repetată a câte unei bile după care bila extrasă este reintrodusă în urnă. Variabila m reprezintă numărul bilelor de o anumită culoare din cele două, în n extrageri succesive, conform schemei cu bila returnată. Numărul p reprezintă proportia de bile de acea culoare în urnă, cu alte cuvinte probabilitatea de obtinere la o extragere simplă a unei bile de culoarea respectivă.

Legea Poisson se referă la o variabilă aleatoare *m* care ia de data aceasta un număr infinit de valori, toate numere naturale. Probabilitătile de aparitie a diferitelor valori se calculează cu relatia

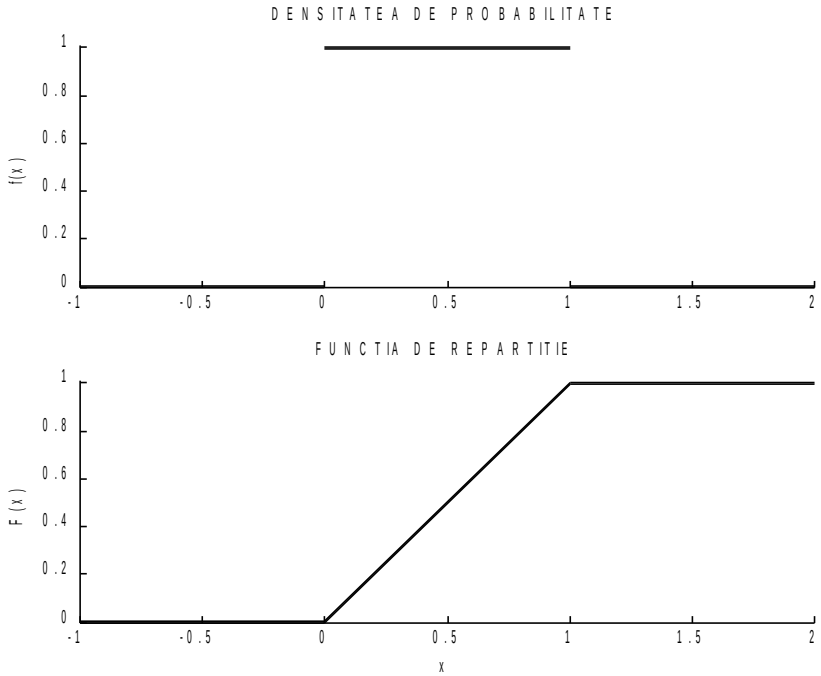
$$P(m) = \frac{\mu^{m}}{m!}\exp(-\mu)$$

în care μ este un parametru real strict pozitiv. Media variabilei este μ, dispersia ei este, de asemenea, μ. Un modelul fizic îl reprezintă numărul dezintegrărilor radioactive, numărul de apeluri telefonice solicitate într-o centrală etc. într-un interval de timp precizat, scurt.

O variabilă aleatoare de tip continuu interesantă este aceea care este *uniform repartizată* pe un inteval finit. Valorile pe care le poate lua o asemenea variabilă au sanse egale de a apărea experimental. Functia de repartitie nu poate fi altfel decât linară: ea este integrala unei functii constante pe intervalul valorilor posibile ale variabilei aleatoare, nule pentru alte valori. Intervalul [*a*, *b*], altfel oarecare dar cu $a < b$ poate fi redus la intervalul standard [0, 1] prin relatia simplă

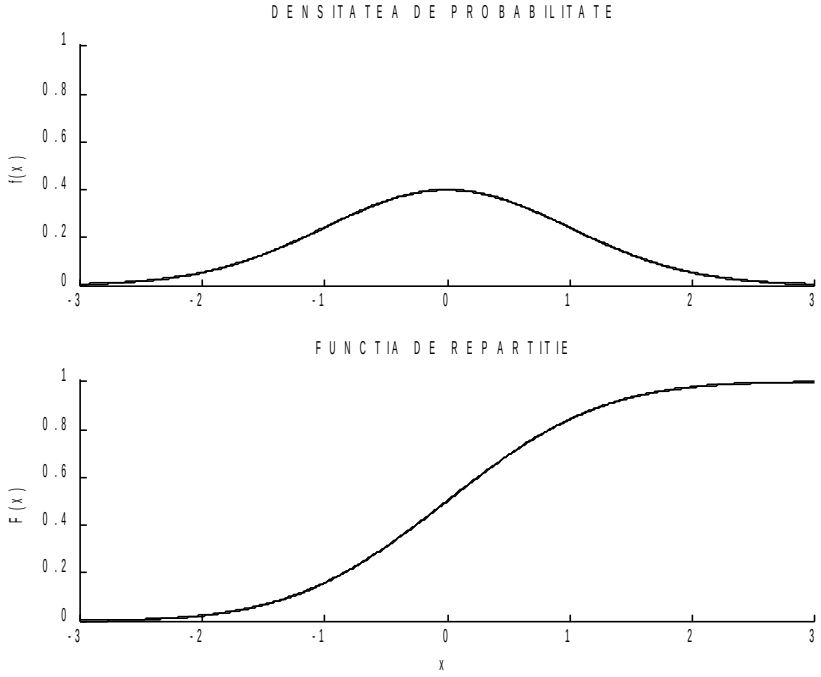
$$x' = (b - x)/(b - a)$$

Trecerea inversă este imediată. Figurile alăturate cuprind graficul functiei densitate de repartitie si graficul functiei de repartitie a unei variabile aleatoare uniform repartizată pe intervalul [0, 1].



Legea de repartitie uniformă

Practic toate limbajele de programare evoluate au implementată sub diferite nume (random, rand etc.) o functie generatoare de numere (pseudo)aleatoare uniform repartizate în intervalul [0, 1]. Pentru simularea fenomenelor aleatoare diverse această functie este de mare utilitate.
O altă lege de repartitie care descrie o variabilă aleatorea de tip continuu este *legea normală* sau *legea gaussiană*. Această lege de repartitie este de importantă fundamentală în calculul probabilităților si în statistica matematică si are multiple aplicatii practice. Aproape ori de câte ori factori întâmplători numerosi actionează asupra unui sistem, actiunea combinată a acestora este percepută prin fenomene cantitative decrise foarte bine de legea normală cunoscută si sub denumirea de legea lui Gauss.



Legea de repartitie normală

Expresia functiei densitate de probabilitate (densitate de repartitie) pentru o variabilă normală (gaussiană) *x* este

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x-m)^2}{2\sigma^2}}$$

Media si dispersia ei apar explicit ca parametrii ai functiei densitate: media este *m*, dispersia este σ^2. În figurile alăturate sunt reprezentate cele două functii pereche, functia densitate de repartitie si functia de repartitie pentru o variabilă normal distribuită, de medie nulă si cu dispersia egală cu unitatea. Această variabilă este numită variabila normală normată. Desi ar putea părea particulară

ea este, dimpotrivă, foarte generală. Orice variabilă normală de medie m si de dispersie σ^2 poate fi redusă la o variabilă normată prin formula simplă

$$z = (x - m)/\sigma$$

După utilizarea variabilei z de medie zero si de dipsersie unitară, tabelată si implementată pe calculatoare, se revine printr-o relatie la fel de simplă la variabila originară x.

Legile de repartitie sunt, desigur, numeroase si modelarea unui fenomen aleator natural cu o lege matematică adecvată reprezintă uneori o problemă.

Verificarea experimentală a legilor de repartitie

Observarea unui fenomen aleator conduce uzual la acumularea unor date experimentale în volum fatalmente finit. Aceste date, să admitem că ele sunt în număr de n, se pot folosi la aprecierea reprezentativităţii unei anumite legi de repartitie pentru fenomenul observat.

Fie t variabila aleatoare observată. Dacă axa t este împărtită în m intervale, valorile observate pot fi sortate si numărate pe fiecare din aceste intervale obtinându-se frecventele n_k si frecventele relative n_k /n pentru fiecare interval I_k (k = 1, 2, …, m).

Dacă se admite că densitatea de repartitie este $f(t)$ atunci se pot calcula probabilităţile

$$p_k = \int_{I_k} f(t)dt$$

asociate fiecărui interval I_k.

Frecventele relative sunt estimări experimentale ale probabilităţilor pentru fiecare din aceste intervale. Se constată, desigur, diferente între probabilităţi si estimările lor. Aceste diferente pot servi la formularea unor ipoteze privind adecvarea modelului teoretic la experimentul observat.

O modalitate de decizie asupra acestei adecvări se bazează pe retinerea diferentei celei mai importante în valoare absolută si compararea ei cu tabele specifice care dau anumite norme în ceea ce priveste abaterea maximă îngăduită. Este vorba aici de utilizarea testului Kolmogorov-Smirnov.

O altă posibilitate mai larg utilizată este aceea care folseste variabila χ^2, o variabilă aleatoare care se prezintă ca o sumă de pătrate ale unor variabile z normale normate independente si este caracterizată de un număr de grade de libertate egal cu numărul termenilor însumati. Matematicienii statisticieni au stabilit că suma

$$\chi^2 = \sum_{k=1}^{m} \frac{(n_k - np_k)^2}{np_k}$$

este într-adevăr o variabilă χ^2 cu m grade de libertate. Valorile de provenientă experimentală sunt comparate cu valori tabelare asociate unor nivele de semnificatie date (uzual 95%). Pentru a accepta ipoteza că o lege de repartitie este reprezentativă pentru variabila aleatoare observată este necesar ca valoarea

χ^2 calculată să fie sub valoarea corespunzătoare nivelului de semnificatie ales, indicată de tabele sau evaluată direct.

Variabile aleatoare multidimensionale

Variabilele aleatoare din expunerea teoretică sau din exemplele prezentate mai sus au fost până acum simple, adică a fost vorba în toate cazurile de o singură aplicatie $X:\Omega \rightarrow R$ legată de un unic câmp de probabilitate (Ω, K, P). Se pot imagina variabile aleatoare cu mai multe componente, variabile sub forma unor vectori cu componente aleatoare definite relativ la un acelasi câmp de probabilitate sau la câmpuri de probabilitate diferite. Astfel legea următoare se referă la o variabilă aleatoare vectorială.

Legea normală n-dimensională dată de densitatea de repartitie

$$f(x) = \frac{1}{(2\pi)^{\frac{n}{2}}\sqrt{\det W}} e^{-\frac{1}{2}(x-m)^T W^{-1}(x-m)}$$

cu media m, un vector cu n componente, si cu matricea de covariatie W, o matrice ($n \times n$) pozitiv definită. Pentru ca exemplul să aibă consistenta necesară trebuie definită mai exact matricea W.

Este de comentat în prealabil problema corelatiei a două variabile aleatoare care pot fi independente, caz în care valorile uneia nu influentează în nici un fel valorile pe care le poate lua cealaltă, dar pot fi mai mult sau mai putin dependente ceea ce înseamnă că dacă una din variabile a luat o valoare atunci legea de repartitie a celeilalte se modifică în functie de acea valoare a primei variabile.

Fiind date două variabile aleatoare x si y de medii nule, media produsului lor $M(xy)$ se numeste *covariatie*. Dacă covariatia este nulă se poate spune în general că cele două variabile nu sunt corelate. Dimpotrivă, dacă $M(xy) \neq 0$ variabilele sunt corelate, există o corelatie între ele, există o dependentă între valorile pe care ele le iau în sensul arătat putin mai devreme. Dacă mediile sunt diferite de zero, afirmatia si definitia se mentin pentru abaterile de la medie. Întrucât covariatia $M(xy)$ poate lua valori foarte diferite, pentru o apreciere cantitativă mai riguroasă a tăriei corelatiei se utilizează coeficientul de corelatie

$$\rho = \frac{M(xy)}{\sqrt{M(x^2)M(y^2)}}$$

care ia valori în intervalul [–1, 1] si în expresia căruia se disting dispersiile celor două variabile, $M(x^2)$ si $M(y^2)$. O valoare pentru ρ apropiată de extremele intervalului indică o corelatie strânsă, o valoare apropiată de zero exprimă o corelatie slabă.

Componentele unui vector aleator, privite ca variabile aleatoare simple sunt mutual mai mult sau mai putin corelate. Se defineste ca matrice a covariatiilor unui vector aleator x media produsului dintre vector si transpusul său $M(xx^T)$. Se obtine o matrice pătrată simetrică care are pe diagonală dispersiile individuale

ale componentelor vectorului aleator. Aceasta este matricea W a covariatiilor utilizată în particular în expresia densitătii de repartitie a variabilei aleatoare normale multidimensionale din exemplul de mai sus. Dacă matricea covariatiilor este diagonală (are toate elementele nule cu exceptia celor de pe diagonala principală) atunci componentele sunt mutual independente. Împărtirea fiecărui element al matricei covariatiilor cu produsul abaterilor medii pătratice ale componentelor vectorului x care corespund pozitiei în matrice produce o matrice a coeficientilor de corelatie, cu 1 pe diagonală, cu valori in intervalul [–1, 1] în rest.

Pentru variabilele aleatoare vectoriale cu componente continue se defineste o functie de repartitie printr-o integrală multiplă, o extindere a integralei din cazul variabilelor simple.

INDICATORI DE FIABILITATE

Dacă T este durata de functionare a unui sistem până la defectare atunci $F(t)$ este notatia pentru functia de repartitie a varibilei aleatoare T si este probabilitatea ca durata de functionare să fie mai mică decât valoarea t.
Complementara probabilitătii de defectare este functia de fiabilitate $R(t)$ care reprezintă probabilitatea ca sistemul să functioneze corect în intervalul $(0, t)$:

$$R(t) = 1 - F(t)$$

Ambele functii se referă la evenimente care se produc în intervalul specificat si nu în momentul t. Ele sunt o notatie mai simplă pentru două functii de interval: $F(0, t)$ si $R(0, t)$.
Pentru un interval oarecare de durată x care începe la momentul t, probabilitatea de defectare este

$$F(t,t+x) = P(t \le T < t+x) = F(t+x) - F(t)$$

si apare ca o probabilitate asociată intervalului $(t, t + x)$ scrisă în conditia certitudinii unei functionări corespunzătoare până la momentul t. Relaxarea absolut necesară a conditiei de certitudine, care oricum nu poate exista, conduce natural la o formulă de probabilitate conditionată

$$F(t,t+x) = P(t \le T < t+x) / P(T \ge t) = [F(t+x) - F(t)] / R(t)$$

si analog, pentru functia de fiabilitate

$$R(t,t+x) = P(T \ge t+x) / P(T \ge t) = R(t+x) / R(t)$$

Functia $R(t, t + x)$ se mai numeste si functia de fiabilitate remanentă.
Functia de distributie $F(t)$ poate avea o derivată

$$f(t) = \lim_{\Delta t \to 0} \frac{F(t+\Delta t) - F(t)}{\Delta t} = \frac{dF(t)}{dt}$$

care este o densitate de probabilitate cu semnificatia de probabilitate de defectare în intervalul $(t, t + \Delta t)$ când întinderea lui tinde către zero. Densitatea de probabilitate dă uzual numele distributiei si dă sens cantitativ probabilitătii de defectare în jurul momentului t.
Pentru descrierea pericolului de defectare în jurul unui moment dat se defineste rata de defectare

$$z(t) = \lim_{\Delta t \to 0} \frac{F(t+\Delta t) - F(t)}{R(t)\Delta t} = \frac{f(t)}{R(t)}$$

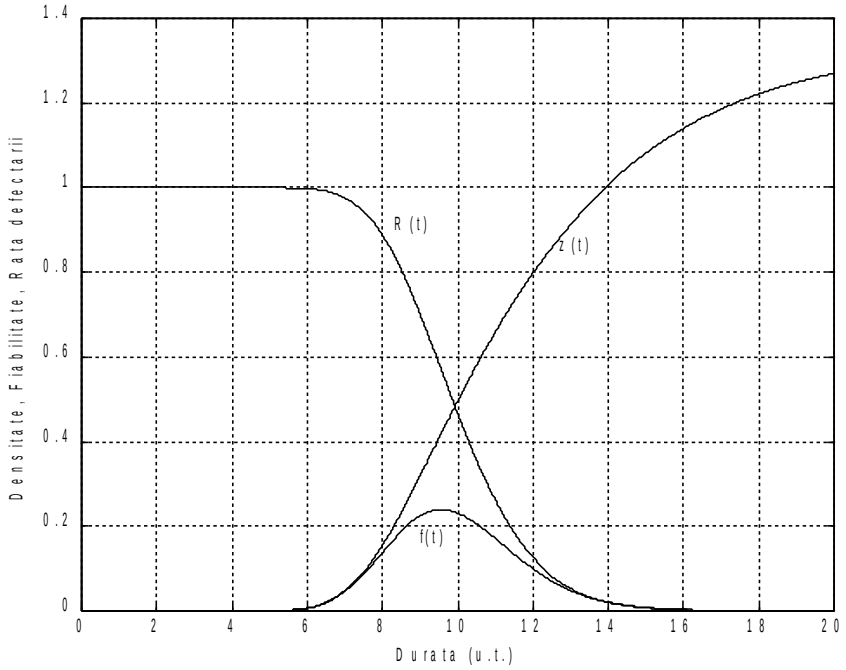
care printr-o înlocuire de-acum familară devine

$$z(t) = -\frac{1}{R(t)}\frac{dR}{dt}$$

Relatia ultimă tratată ca o ecuatie diferentială si integrată conduce la

$$R(t) = e^{-\int_0^t z(u)du}$$

relatie de mare importantă între indicatorii de fiabilitate.



Media timpului de functionare este

$$m = \int_0^\infty tf(t)dt$$

si după o integrare prin părti

$$m = \int_0^\infty R(t)dt$$

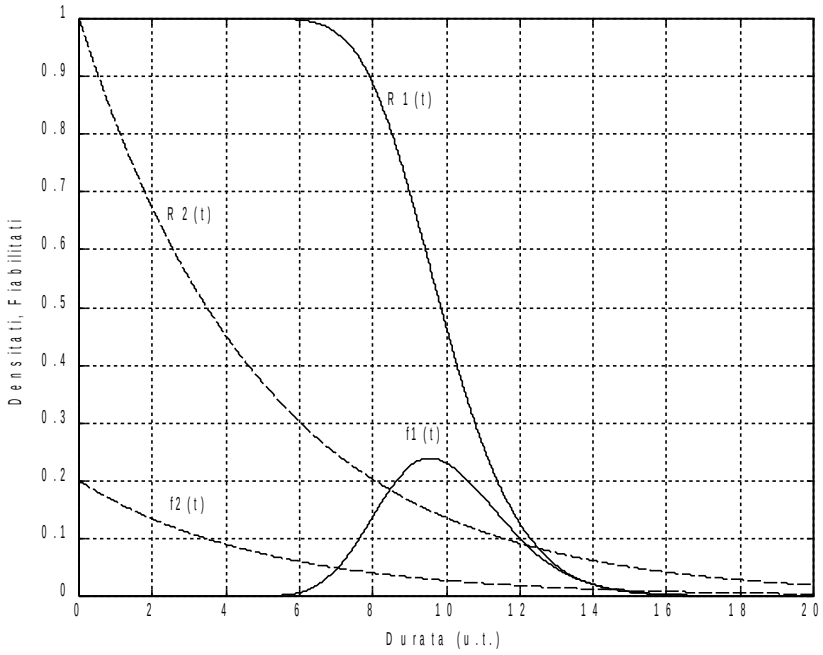
Aceasta este media timpului până la defectare (Mean Time To Failure – MTTF). Defectarea este presupusă unică. În cazul readucerii (repetate) a sistemului la parametrii initiali, după fiecare defectare se poate vorbi de timpul mediu între două defectări succesive (Mean Time Between Failure – MTBF). În cazul readucerii sistemului într-o stare diferită de cea initială media *m* se referă la timpul mediu până la prima defectare (Mean Time To First Failure – MTTFF). S-au dat aici si denumirile în limba engleză si prescurtările lor deoarece în multe lucrări din domeniu atât denumirile cât si prescurtările sunt utilizate ca atare.
O altă medie importantă este

$$m(t) = \int_0^{\infty} R(t, t + x)dx = \frac{1}{R(t)} \int_t^{\infty} R(u)du$$

Aceasta este media timpului de functionare rămas până la defectarea unui sistem. Pentru $t = 0$, media ultimă coincide cu media din relatia anterioară.
Se calculează uneori si o dispersie a timpului de functionare

$$D = \int_0^{\infty} (t - m)^2 f(t)dt; \quad \sigma = \sqrt{D}$$

Această dispersie măsoară gradul de uniformitate a performantelor unor sisteme identice. O tenhologie bine pusă la punct în productia acelor sisteme conduce la dispersii mici.
Se pot defini, de asemenea, cvantile ale timpului de functionare ca solutii ale ecuatiei $F(t_\alpha) = \alpha$ cu α o probabilitate specificată, legată de cele mai multe ori de un timp de garantie.



În evaluarea a două sisteme sub aspectul fiabilitătii se compară mai multi parametri, în raport cu situatia concretă. În figura alăturată sistemul 1 este potrivit pentru o durată de utilizare limitată, inferioară celei care corespunde punctului de intersectie a graficelor pentru fiabilităti; sistemul 2 este potrivit unei misiuni tehnologice nedefinite ca durată. Se mai pot compara mediile timpilor de functionare până la prima defectare si alte valori caracteristice.

Uzura

Uzura este un fenomen fizic prezent în foarte multe sisteme de tipul *hardware*. Sistemele *software* sunt fără uzură. Pentru acestea nu există un echivalent al uzurii fizice din cazul sistemelor *hardware*. Uzura morală a unor programe despre care se vorbeste adesea nu intră în preocupările prezentului curs.

Uzura (fizică) este calificată drept pozitivă dacă functia de fiabilitate $R(t, t + x)$ este descrescătoare cu t pe intervalul $(0, \infty)$ pentru orice $x \geq 0$. Asadar, pentru un sistem cu uzură pozitivă fiabilitatea scade cu cresterea vârstei. Invers, sistemul este cu uzură negativă dacă functia $R(t, t + x)$ este crescătoare cu t pentru orice $x \geq 0$.

Transferând definitia în termeni de rată de defectare, care se poate scrie si sub forma

$$z(t) = \lim_{x \to 0} \frac{R(t) - R(t + x)}{xR(t)} = \lim_{x \to 0} \frac{R(t, t + x)}{x}$$

se consideră că sistemul este cu uzură pozitivă dacă rata de defectare $z(t)$ este o functie crescătoare si cu uzură negativă dacă $z(t)$ este descrescătoare. Invers, din relatia

$$R(t, t + x) = e^{-\int_t^{t+x} z(u)du}$$

se deduce relatia de crestere/descrestere a functiei de fiabilitate si a ratei de defectare în cazurile uzurii pozitive sau negative.

Uzura medie. Se numeste sistem cu uzură medie pozitivă un sistem pentru care functia $(1/t)\ln(1/R)$ este crescătoare în timp. Functia are o scriere echivalentă, $(1/t)\int_0^t z(u)du$, care are în vedere rata defectărilor. Sistemele cu functia mentionată descrescătoare sunt sisteme cu uzură medie negativă. Sub aspect practic, sistemele cu uzură pozitivă sunt afectate de un proces de deteriorare functională progresivă, iar sistemele cu uzură negativă functionează din ce în ce mai bine pe măsura trecerii timpului. Lipsa de variatie, constanţa functiei $(1/t)\ln(1/R)$ echivalează cu lipsa uzurii.

Uzura pozitivă nu face defectarea sistemului iminentă la un anumit moment cum nici uzura negativă nu exclude defectarea până la un anumit moment. Fenomenul defectării rămâne aleator, imprevizibil, numai parametrii legilor de repartitie se modifică diferit într-un caz sau în celălalt.

În lucrările de specialitate în limba enlgeză sau în alte limbi, pentru a califica un sistem din punct de vedere al uzurii se utilizează unele prescurtări mai mult sau mai putin consacrate care sunt reproduse imediat:

IFRA – Increasing Failure Rate Average – medie a ratei de defectare crescătoare

DFRA – Decreasing Failure Rate Average – medie a ratei de defectare descrescătoare

IFR – Increasing Failure Rate – rată de defectare crescătoare

DFR – Decreasing Failure Rate – rată de defectare descrescătoare

Aceste calificative sunt în relatia IFR$\Rightarrow$ IFRA, DFR$\Rightarrow$ DFRA. Implicatiile nu sunt valabile si invers.

Demonstratie. Dacă un sistem este IFR atunci functia $\ln[1/R(t)]$ este convexă

$$\ln\frac{1}{R(t)} = \int_0^t z(u)du \Rightarrow \left(\ln\frac{1}{R(t)}\right)' = z(t) \Rightarrow \left(\ln\frac{1}{R(t)}\right)'' = z'(t) > 0$$

derivata unei functii convexe pozitive care trece prin origine este crescătoare pe intervalul $(0, \infty)$ si atunci se poate afirma că functia

$$\frac{\ln\frac{1}{R(t)} - \ln\frac{1}{R(0)}}{t}$$

este crescătoare pe acelasi interval. Acelasi mers al demonstratiei pentru DFR tinând seama de concavitatea functiei $\ln[1/R(t)]$.

În ipoteza că functia $\ln[1/R(t)]$ nu este derivabilă, de pildă în cazul sistemelor cu rată de defectare constantă pe portiuni, implicatiile de mai sus rămân valabile.

Sistemele cu degradare sunt sisteme pentru care functia de fiabilitate relativă la o utilizare de durată x care începe la momentul t este mai mică decât functia de fiabilitate pentru intervalul $(0, x)$ oricare ar fi vârsta t si oricare ar fi durata x

$$R(t, t+x) < R(x), \quad t, x > 0$$

cu alte cuvinte, după utilizare un asemenea sistem este inferior unuia nou (NBU – New Better than Used. Un sistem IFR este NBU dar nu totdeauna si invers (exercitiu).

Se poate vorbi si de sisteme fără degradare (NWU – New Worse than Used), pentru care

$$R(t, t+x) > R(x), \quad t, x > 0$$

cu implicatia DFR$\Rightarrow$ NWU dar nu si reciproc.

Asadar calitătile NBU (NWU) sunt mai generale decât IFR (DFR) si chiar decât IFRA (DFRA). Ultima afirmatie se sustine prin demonstratia care urmează.

Prin definitie un sistem IFRA are functia $(1/t)\ln(1/R)$ crescătoare si, implicit, $[1/R]^{1/t}$ crescătoare si $[R(t)]^{1/t}$ descrescătoare. Se poate scrie atunci

$$[R(t+x)]^{\frac{1}{t+x}} < [R(t)]^{\frac{1}{t}}$$

si apoi

$$R(t, t+x) < [R(t)]^{\frac{x}{t}}$$

deoarece $R(t, t+x) = R(t+x)/R(t)$.

Pentru un $x \in [0, t]$ are loc

$$[R(t)]^{\frac{1}{t}} \le [R(x)]^{\frac{1}{x}}$$

si apoi, tinând seamă de relatia anterioară, rezultă

$$R(t, t+x) < [R(x)]^{\frac{1}{x}\cdot x} = R(x)$$

ceea ce exprimă calitatea de NBU a sistemului.
Fie acum $x > t$. Cum functia $[R(t)]^{1/t}$ este descrescătoare, se poate scrie

$$[R(t)]^{\frac{1}{t}} \geq [R(x)]^{\frac{1}{x}}$$

si apoi din relatiile de mai sus se poate scrie iarăsi

$$[R(t+x)]^{\frac{1}{t+x}} < [R(x)]^{\frac{1}{x}}$$

si mai departe

$$R(x, x+t) < [R(x)]^{\frac{t}{x}}$$

dar, de data aceasta

$$[R(x)]^{\frac{1}{x}} \leq [R(t)]^{\frac{1}{t}}$$

de unde rezultă

$$R(x, x+t) < R(t)$$

ceea ce înseamnă, din nou, că sistemul este NBU.
Sisteme cu degradare în medie. Un sistem cu degradare în medie este un sistem care are media timpului de functionare rămas mai mică decât media timpului de functionare a sistemului, $m(t) < m$ (NBUE – New Better than Used in Expectation). Relatia din definitie este, în dezvoltare, tot una cu

$$\int_0^\infty R(t, t+x)dx < \int_0^\infty R(x)dx$$

sau după mici modificări

$$\int_t^\infty R(x)dx < mR(t)$$

Implicatia NBU $\Rightarrow$ NBUE este acum evidentă.
Există sisteme care nu pot fi încadrate în nici una din categoriile mentionate. Acestea sunt sistemele fără uzură care se exprimă în termeni de fiabilitate astfel

$$R(t, t+x) = R(x), \quad t, x \geq 0$$

Pentru sistemele de acest gen functia de repartitie a duratelor de viată, altfel spus a duratelor de functionare până la (prima) defectare este $F(t) = 1 - e^{-\lambda t}$, densitatea repartitiei lor este $f(t) = \lambda e^{-\lambda t}$, rata defectărilor este constantă $z(t) = \lambda$, iar media si dispersia duratelor de viată sunt $m(t) = m = 1/\lambda$, respectiv $\sigma^2 = 1/\lambda^2$. Una din aplicatiile importante ale acestui model se referă la sistemele *software*.

Legi de repartitie utilizate in teoria fiabilitătii sistemelor

Pentru sistemele fără reînnoire, adică pentru acele sisteme care odată defecte sunt iremediabil defecte, durata lor de viată este o variabilă aleatoare.

Nume	$f(t)$	Media	Dispersia
Gamma	$\frac{\beta\,(\beta\,t)^{\alpha-1}e^{-\beta\,t}}{\Gamma\,(\alpha)}$	$\frac{\alpha}{\beta}$	$\frac{\alpha}{\beta^{\,2}}$
Weibull	$\frac{\beta\,t^{\beta-1}}{\alpha}e^{-\frac{t^{\beta}}{\alpha}}$	$\alpha^{\frac{1}{\beta}}\Gamma\left(1+\frac{1}{\beta}\right)$	$\alpha^{\frac{2}{\beta}}\left[\Gamma\left(1+\frac{2}{\beta}\right)-\right.$ $\left.-\,\Gamma^{\,2}\left(1+\frac{1}{\beta}\right)\right]$
Normală	$\frac{1}{\sqrt{2\pi}\,\sigma}e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^{2}}$	μ	$\sigma^{\,2}$
Lognormală	$\frac{1}{\sqrt{2\pi}\,\sigma\,t}e^{-\frac{1}{2}\left(\frac{\ln t-\mu}{\sigma}\right)^{2}}$	$e^{\mu+\frac{\sigma^{\,2}}{2}}$	$e^{2\mu+\sigma^{\,2}}\left(e^{\sigma^{\,2}}-1\right)$
A valorii extreme (Gumbel)	$e^{\frac{t-\mu}{\eta}}e^{-e^{\frac{t-\mu}{\eta}}}/\left(\eta\,e^{-e^{\frac{-\mu}{\eta}}}\right)$		
Rayleigh	$\frac{2}{\omega^{\,2}}te^{-\frac{t^{2}}{\omega^{\,2}}}$	$\omega\,\Gamma\left(\frac{3}{2}\right)$	$\omega^{\,2}\left[1-\Gamma^{\,2}\left(\frac{3}{2}\right)\right]$
Rayleigh generalizat	$\frac{2\theta^{\,k+1}}{\Gamma\,(k+1)}t^{2k+1}e^{-\theta\,t^{2}}$	$\frac{\Gamma\left(k+\frac{3}{2}\right)}{\Gamma\,(k+1)}\frac{1}{\sqrt{\theta}}$	$\left[k+1-\frac{\Gamma^{\,2}\left(k+\frac{3}{2}\right)}{\Gamma^{\,2}(k+1)}\right]\frac{1}{\theta}$
Alfa	$\frac{\beta}{t^{2}\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{\beta}{t}-\alpha\right)^{2}}$	$\frac{\beta}{\alpha}\left(1+\frac{1}{\alpha^{\,2}}\right)$	$\frac{\beta^{\,2}}{\alpha^{\,4}}\left(1+\frac{8}{\alpha^{\,2}}\right)$
Putere	$\delta\,b^{-\delta}t^{\delta-1}\quad t\in(0,b)$	$\frac{\delta\,b}{\delta+1}$	$\frac{\delta\,b^{2}}{(\delta+2)(\delta+1)^{2}}$
Birnbaum-Saunders	$\frac{1}{2\sqrt{2\pi}\,\alpha\,t}\cdot$ $\left(\sqrt{\frac{t}{\beta}}+\sqrt{\frac{\beta}{t}}\right)\cdot$ $e^{-\frac{1}{2\alpha^{\,2}}\left(\frac{t}{\beta}+\frac{\beta}{t}-2\right)}$	$\beta\left(1+\frac{\alpha^{\,2}}{2}\right)$	$(\alpha\,\beta)^{2}\left(1+\frac{5\alpha^{\,2}}{4}\right)$

O variabilă aleatoare este complet definită de functia ei de repartitie. Durata de viată este o variabilă aleatoare de tip continuu, prin urmare functia de repartitie este o functie continuă si derivabilă în raport cu timpul până la prima (si ultima)

defectare. În cazul continuității variabilei, densitatea ei de repartitie este de asemenea capabilă să o descrie complet. Tabelul alăturat contine un număr de densități de repartitie a duratei de viată a sistemelor, foarte frecvent utilizate si confirmate de practică. Legile de repartitie cuprinse în tabel sunt departe de a fi acoperitoare pentru toate situatiile practice posibile.

Aproximări ale functiilor densitate de repartitie prin exponentiale

Aproximarea este necesară ori de câte ori nici una din densitătile de repartitie consacrate nu se potriveste unei anumite experiente privind fiabilitatea unui sistem. Aproximarea se poate realiza în trei moduri, conform modelelor *în serie*, *în paralel* sau *în triunghi*.
Aproximarea *serie* constă într-o combinatie liniară de exponentiale

$$f(t) = \sum_{i=1}^{n} \omega_i \lambda_i e^{-\lambda_i t}$$

cu coeficientii ω_i îndeplinind conditia $\sum_{i=1}^{n} \omega_i = 1$, adică combinatia liniară este convexă.
Interpretarea fizică a combinatiei este aceea a unui sistem cu mai multe moduri de defectare, mutual incompatibile. Exemplele practice le furnizează sistemele a căror defectare poate consta într-un scurtcircuit sau într-o întrerupere. Fiecare din modalitătile de defectare respectă o lege probabilistică exponentială, are o probabilitate de producere ω_i si are un parametru λ_i specific.
Aproximarea *paralel* are în vedere un timp de functionare până la defectare, care se prezintă ca o sumă de durate aleatoare independente una de alta, fiecare cu o lege exponentială de parametru λ_i. Compunerea densitătilor de repartitie a două din aceste durate statistic independente se face prin operatia de convolutie

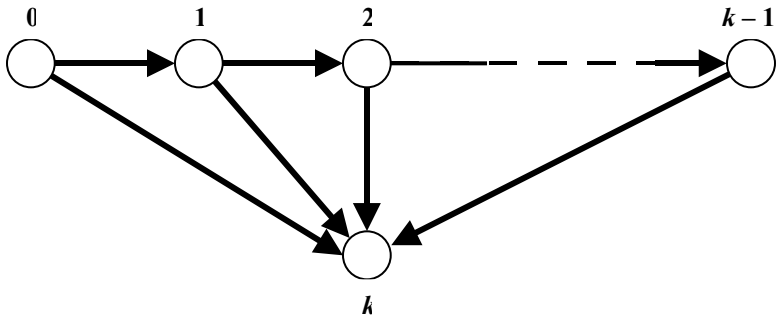
$$f_{ij}(t) = (f_i * f_j)(t) = \int_{-\infty}^{+\infty} f_i(\tau) f_j(t - \tau) d\tau$$

operatie care se poate repeta prin adăugarea altor functii până la completa lor epuizare. Dacă se notează mai simplu functia rezultată cu $f(t)$ atunci, cu transformarea Laplace, se obtine

$$f^*(s) = L[f(t)] = \prod_{i=1}^{n} \frac{\lambda_i}{s + \lambda_i}$$

Dacă parametrii λ_i sunt distincti atunci prin transformarea Laplace inversă se obtine o combinatie liniară de exponentiale asemănătoare celei de la aproximarea serie. Dacă parametrii λ_i sunt si cu repetitie se obtin repartitii Γ de ordine întregi diferite. În cazul în care toti λ_i au aceeasi valoare se obtine o repartitie Γ unică, de ordinul întreg $n - 1$.
Aproximarea *triunghi* se potriveste sistemelor care trec prin mai multe stări intermediare (ca sistemele în paralel) dar nu trebuie să le parcurgă obligatoriu pe toate. Defectarea care scoate din functiune sistemul se poate produce în orice

moment, indiferent de starea curentă a sistemului. Descrierea este foarte sugestivă printr-un graf al tranzitiilor. Alăturat este dat un astfel de graf.



Tranzitiile posibile într-un interval Δt sunt $(0\rightarrow 1)$, $(1\rightarrow 2)$, ..., $(k-2\rightarrow k-1)$ dar si $(0\rightarrow k)$, $(1\rightarrow k)$, $(2\rightarrow k)$, ..., $(k-1\rightarrow k)$ pe lângă stationarea în nodul curent 0, 1, 2, ..., $k-1$. Se întelege că starea k este starea de nefunctionare.
Probabilitătile asociate tranzitiilor sunt proportionale cu intervalul Δt si sunt respectiv $\lambda_1\Delta t$, $\lambda_2\Delta t$, …, $\lambda_{k-1}\Delta t$, apoi $\lambda_k\Delta t$, $\lambda_{k+1}\Delta t$, …, $\lambda_{2k-1}\Delta t$ si încă $1-(\lambda_{i+1}+\lambda_{k+1})\Delta t$ pentru persistenta sistemului în starea i. Trecerea dintr-o stare în alta are toate caracteristicile unui proces Markov. Probabilitătile stărilor la un moment dat sunt

$$p_i(t) = \sum_{j=1}^{i} d_{ij} \exp[-(\lambda_j + \lambda_{k+j})t]$$

cu

$$d_{ij} = \prod_{l=1}^{i-1} \frac{\lambda_l}{\prod_{\substack{l=1 \\ l\neq j}}^{i} (\lambda_i - \lambda_j + \lambda_{k+i} - \lambda_{k+j})}$$

Combinatiile de exponentiale în schemele serie, paralel, triunghi oferă o multitudine de posibilităti de aproximare a fiabilitătii sistemelor reale. Grafurile asociate sunt desigur mai complicate. Oricare ar fi modelul ales el trebuie trecut printr-un test de concordantă cu date experimentale.

Aproximarea discretă

Aproximarea discretă se realizează prin juxtapunerea pe intervale adecvate a unor legi exponentiale. Pe fiecare interval uzura este considerată constantă. Functia de fiabilitate este în cazul aproximării

$$R^*(t) = \begin{cases} \exp\left\{-\left[\sum_{j=1}^{i-1} \lambda_j (t_j - t_{j+1}) + \lambda_i (t - t_i)\right]\right\} & t \in [t_{i-1}, t_i] \\ 0 & t > t_i \end{cases}$$

Demonstrarea faptului că orice lege reală de repartitie poate fi aproximată discret este facilă.
Operatia de punere în acord a modelului cu realitatea este cunoscută sub numele de *estimare de parametri*. Cu cât sunt în joc mai multi parametri cu atât problema este mai complicată.
Una din metodele de estimare a parametrilor este cea a *verosimilitătii maxime*. Pentru a utiliza această metodă se construieste functia de verosimilitate $L(X/\Theta)$ care contine vectorul X al observatiilor experimentale si vectorul Θ al parametrilor de estimat. Încercările experimentale pot fi trunchiate (sau cenzurate, cum se mai spune), cu sau fără înlocuire. Fie o încercare cenzurată, fără înlocuire, si o lege exponentială. Atunci

$$L(t_1,t_2,...,t_r/\lambda) = A_n^r(\lambda e^{-\lambda t_1})(\lambda e^{-\lambda t_2})...(\lambda e^{-\lambda t_r})(e^{-\lambda t_r})^{n-r} =$$

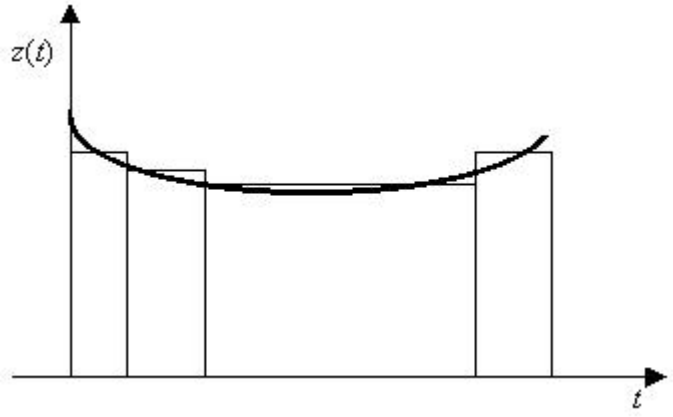
$$= A_n^r \lambda^r e^{-\lambda[\sum t_i + (n-r)t_r]} = A_n^r \lambda^r e^{-\lambda T_\Sigma}$$

cu

$$T_\Sigma = \sum_{i=1}^{r} t_i + (n-r)t_r$$

Functia de verosimilitate este maximă pentru valoarea λ care anulează derivata $\frac{d}{d\lambda} L(T_\Sigma/\lambda)$, adică verifică ecuatia

$$A_n^r r\lambda^{r-1} e^{-\lambda T_\Sigma} - A_n^r T_\Sigma \lambda^r e^{-\lambda T_\Sigma} = 0$$



ceea ce conduce la $\frac{r}{\hat{\lambda}} = T_\Sigma$ si apoi la $\hat{\lambda} = \frac{r}{T_\Sigma}$, care este estimatia maxim verosimilă.
Trebuie verificat dacă estimatia este nedeplasată sau, cum se mai spune, este absolut corectă, adică are proprietatea

$$\int_{-\infty}^{+\infty} \hat{\Theta}\, g(\hat{\Theta}/\Theta)\, d\hat{\Theta} = \Theta$$

Integrala este o medie a variabilei aleatoare $\hat{\theta}$, cu luarea în considerare a densitătii de repartitie conditionată $g(\hat{\theta} / \theta)$.

În cazul parametrului unic λ de mai sus, se scrie mai întâi diferit variabila T_Σ

$$T_\Sigma = \sum_{i=1}^{r} t_i + (n - r)t_r =$$

$$= nt_1 + (n-1)(t_2 - t_1) + (n-2)(t_3 - t_2) + ... + (n - r + 1)(t_r - t_{r-1})$$

Se notează $s_k = (n - k + 1)(t_k - t_{k-1})$ ceea ce face ca

$$T_\Sigma = \sum_{i=1}^{r} s_i$$

Vectorul aleator $s^T = [s_1 s_2 ... s_r]$ are densitatea de repartitie

$$L(s_1, s_2, ..., s_r) = \frac{L(t_1, t_2, ..., t_r)}{\begin{vmatrix} \frac{\partial s_1}{\partial t_1} & \cdots & \frac{\partial s_1}{\partial t_r} \\ \cdots & \cdots & \cdots \\ \frac{\partial s_r}{\partial t_1} & \cdots & \frac{\partial s_r}{\partial t_r} \end{vmatrix}} = \frac{A_n^r \lambda^r e^{-\lambda T_\Sigma}}{\begin{vmatrix} n & 0 & \cdots & 0 \\ -n+1 & n-1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & n-r+1 \end{vmatrix}}$$

cu determinantul functional bidiagonal (toate elementele de deasupra diagonalei principale si toate elementele de sub diagonala a doua sunt nule). Fiecare din variabilele s_i este independentă de celelalte si este repartizată exponential cu acelasi parametru λ . Rezultă

$$g(T_\Sigma / \lambda) = \frac{\lambda (\lambda T_\Sigma)^{r-1} e^{-\lambda T_\Sigma}}{\Gamma(r)}$$

cu *r* determinat. Media parametrului estimat este

$$M(\hat{\lambda}) = \int_0^\infty \frac{r}{T_\Sigma} \frac{\lambda (\lambda T_\Sigma)^{r-1} e^{-\lambda T_\Sigma}}{(r-1)!} dT_\Sigma = \frac{r\lambda^r}{(r-1)!} \int_0^\infty T_\Sigma^{r-2} e^{-\lambda T_\Sigma} dT_\Sigma = \frac{r\lambda^r (r-2)!}{(r-1)!\lambda^{r-1}} = \frac{r\lambda}{r-1}$$

ceea ce indică o estimatie deplasată. Estimatia nedeplasată este

$$\hat{\lambda} = \frac{r-1}{T_\Sigma}$$

SISTEME CU SCHIMBARE

Reînnoirea ca proces aleator

Se presupune că sistemele tratate în această sectiune pot fi readuse în stare de functionare de îndată ce se constată o defectiune care le scoate din functiune. Se admit interventii de foarte scurtă durată, practic neglijabilă. O interventie înseamnă o reînnoire. Evolutia sistemului este marcată de momentele de reînnoire t_1, t_2, …, t_n si de intervalele între reînnoiri X_1, X_2, …, X_n. Numărul de reînnoiri N_T petrecute în intervalul $(0,t)$ este un proces aleator discret. În ceea ce priveste variabilele X_i (i = 1, 2, …, n), apare ratională acceptarea ipotezei independentei lor. Acest fapt permite tratarea în fiecare interval a acelorasi indicatori de fiabilitate. În realitate schimbările pot influenta fiabilitatea sistemului, uneori în bine, alteori în rău.

Fie $R_i(x)$ functia de fiabiliate pe inervalul (t_{i-1}, t_i). Reînnoirile se clasifică după relatia între functiile $R_i(x)$.

Reînnoire propriu-zisă este o reînnoire care aduce sistemul în aceeasi stare de dinaintea defectării, $R_i(x) = R(x)$ pentru orice i. Se mai numeste proces de reînnoire simplu. Alte tipuri de reînnoiri sunt cazuri mai generale. De pildă, dacă $R_1(x) > R_2(x) > \ldots > R_n(x)$ reînnoirile sunt pozitive, iar dacă $R_1(x) < R_2(x) < \ldots < R_n(x)$ reînnoirile sunt negative.

Sunt interesante sub aspect practic reînnoirile fără modificarea de la o reînnoire la alta a gradului de uzură. Reînnoirea în aceste cazuri poate fi pozitivă sau negativă, după tipul de uzură a sistemului. La un sistem fără uzură reînnoirea este simplă.

Procesul aleator N_T are descrierea matematică prezentată mai jos.

Fie intervalul $(0, t)$, intervalul relativ scurt $(t, t + \Delta t)$ si probabilitatea ca până la momentul $t + \Delta t$ să se fi produs r reînnoiri, $P_r(t + \Delta t) = P(N_{t+\Delta t} = r)$. Cele r reînnoiri se pot produce în mai multe moduri, conform tabelului care urmează.

În intervalul $(0, t)$	În intervalul $(t, t + \Delta t)$
r	0
r-1	1
r-2	2
...	...
0	r

La un sistem fără uzură, probabilitatea defectării în intervalul $(t, t + \Delta t)$ este $\lambda \Delta t + O[(\Delta t)^2]$, este asadar, exceptând un infinit mic de ordin superior lui Δt,

proportională cu durata (scurtă) Δt. Defectările multiple în intervalul Δt sunt în aceste conditii practic excluse. Se poate scrie relatia de recurentă

$$P_r(t+\Delta t) = P_{r-1}(t)\{\lambda t + O[(\Delta t)^2]\} + P_r(t)\{1 - \lambda t - O[(\Delta t)^2]\}$$

care prin trecere la limită $(\Delta t \to 0)$ conduce la

$$\frac{dP_r(t)}{dt} = -\lambda P_r(t) + \lambda P_{r-1}(t)$$

Aceasta este o ecuatie diferentială care se integrează după schimbarea de functie $P_r(t) = v_r(t)e^{-\lambda t}$ cu rezultatul integrării

$$P_r(t) = \frac{(\lambda t)^r}{r!}e^{-\lambda t}$$

care este exact legea de repartitie Poisson, modelul adecvat pentru cel mai simplu proces de reînnoire.

Media numărului de reînnoiri în intervalul (0, t) poartă numele de functie de reînnoire

$$H(t) = M(N_t) = \lambda t$$

Se defineste de asemenea, densitatea de reînnoire

$$h(t) = \frac{dH(t)}{dt} = \lambda$$

si o dispersie a reînnoirilor

$$D(t) = M(N_t^2) - H^2(t) = (\lambda t)^2$$

care este dependentă de timp.

Pentru sistemele cu uzură evaluările sunt mai complicate dar nu extrem de complicate. Există relatia

$$H(t) = \int_0^t z(t)dt$$

care leagă functia de reînnoire de rata de defectare. Rezultă imediat că $h(t) = z(t)$. Si pentru că

$$H(t) = \ln\frac{1}{R(t)}$$

pentru un interval (t_1, t_2) se obtine

$$H(t_1,t_2) = H(t_2) - H(t_1) = \ln\frac{1}{R(t_2)} - \ln\frac{1}{R(t_1)} = \ln\frac{1}{R(t_1,t_2)}$$

Cazul general cere efectuarea unei distinctii între tipurile de înlocuire (propriu-zisă, negativă sau pozitivă). Pentru a distinge între tipurile de reînnoire, pe baza unor intervale între reînnoiri consecutive, uzual primele trei, se formulează o ipoteză de nul de genul

$$H_0: R_{i-1}(x) = R_i(x) = R_{i+1}(x)$$

si alternativa

$$H_1: R_{i-1}(x) > R_i(x) > R_{i+1}(x)$$

Dacă $x_{i_1}, x_{i_2}, x_{i_3}$ sunt duratele de functionare până la a treia înlocuire corespunzătoare sistemului i din esantionul ($i = 1, 2, ..., n$) atunci functiile

$$Z_i = \begin{cases} 1 & \text{pentru } x_{i_1} < x_{i_2} < x_{i_3} \\ 0 & \text{altminteri} \end{cases}$$

sunt prin valorile lor argumente împotriva ($Z_i = 1$) sau în favoarea ($Z_i = 0$) caracterului pozitiv al reînnoirilor. Discriminarea se face prin suma

$$S = \sum_{i=1}^{n} Z_i$$

în raport cu o valoare-criteriu k citită în tabele specializate sau calculată dar corespunzătoare unei valori α asociată riscului de a respinge ipoteza H_0 când ea este de fapt corectă. Dacă valoarea k majorează valoarea calculată S atunci se acceptă ca valabilă ipoteza alternativă H_1. Riscul α se asociază asadar probabilitătii

$$\alpha = P(S \le k) = \sum_{r=0}^{k} C_n^r p_0^r (1 - p_0)^{n-r}$$

cu $p_0 = P(Z_i = 1 / H_0) = 1/6$ în cazul reînnoirilor propriu-zise, ceea ce face din relatia de mai sus o ecuatie în k.

Dacă sistemul este de tipul cu reînnoire propriu-zisă, asadar este readus mereu la starea din momentul $t = 0$, atunci

$$P(N_t \ge r) = P(T_r < t)$$

Numărul de reînnoiri produse în intervalul $(0, t)$ este mai mare decât r dacă si numai dacă durata T_r scursă până la reînnoirea cu numărul r este inferioară lui t.

Se notează cu $K_r(t)$ functia de repartitie a duratei T_r si cu $k_r(t)$ densitatea ei de repartitie. Procesul aleator N_t poate fi exprimat cu ajutorul acestor functii

$$P(N_t = r) = P(N_t \ge r) - P(N_t \ge r + 1) = K_r(t) - K_{r+1}(t)$$

cu $r = 1, 2, ...$ si $K_0(t) = 1$.

Variabila $T_r = X_1 + X_2 + \ldots + X_r$ are o densitate de repartitie care se calculează pe baza unor intervale între reînnoiri consecutive, uzual primele trei cu relatia $k_r(t) = f(t) \otimes f(t) \otimes ... \otimes f(t)$, o convolutie multiplă de r factori identici. Prin transformarea Laplace rezultă $k_r^*(s) = [f^*(s)]^r$ si $K_r^*(s) = (1/s)[f^*(s)]^r$. În cazul unui proces de reînnoire general, densitatea de repartitie pe primul interval este $f_1(t)$, diferită de densitatea $f(t)$ pentru intervalele următoare. Atunci avem pentru $k_r^*(s) = f_1^*(s)[f^*(s)]^{r-1}$ si pentru $K_r^*(s) = (1/s) f_1^*(s)[f^*(s)]^{r-1}$.

Functia de reînnoire este

$$H(t) = \sum_{r=1}^{\infty} rP(N_t = r) = \sum_{r=1}^{\infty} r[K_r(t) - K_{r+1}(t)] = \sum_{r=1}^{\infty} K_r(t)$$

si

$$h(t) = \frac{dH(t)}{dt} = \sum_{r=1}^{\infty} k_r(t)$$

În domeniul Laplace, pentru reînnoirea simplă

$$h^*(s) = \sum_{r=1}^{\infty} [f^*(s)]^r = \frac{f^*(s)}{1 - f^*(s)}$$

si

$$H^*(s) = \frac{f^*(s)}{s[1 - f^*(s)]}$$

Pentru cazul general

$$h^*(s) = \sum_{r=1}^{\infty} f_1^*(s)[f^*(s)]^{r-1} = \frac{f_1^*(s)}{1 - f^*(s)}$$

$$H^*(s) = \frac{f_1^*(s)}{s[1 - f^*(s)]}$$

care poate redeveni modelul procesului simplu prin substituirea functiilor $f(t), f^*(s)$ în loc de $f_1(t), f_1^*(s)$.

În general

$$h(t) = f_1(t) + h(t) \otimes f(t) = f_1(t) + \int_0^t h(\tau) f(t - \tau) d\tau$$

relatie cunoscută si ca *ecuatia reînnoirii*. În jurul momentului *t* se poate produce prima reînnoire cu o probabilitate exprimată de $f_1(t)$. Dacă reînnoirea anterioară s-a produs la momentul τ, o reînnoire de un ordin oarecare produsă în jurul momentului *t* are sansa/probabilitatea de a se produce exprimată de integrala de convolutie.

Disponibilitatea sistemelor

Disponibilitatea se referă la sistemele cu reînnoire pentru care durata reînnoirii încetează să mai fie neglijabilă. Mai mult, este aleatoare si este ea însăsi descrisă de o functie de repartitie a timpului în care se realizează reînnoirea, de o densitate de repartitie a duratei reînnoirii asociată cu probabilitatea ca sistemul să fie pus în functiune în jurul momentului *t*. Probabilitatea punerii în functiune în jurul acelui moment conditionată de neîncheierea reînnoirii la acel moment

$$z_2(t) = \frac{f_2(t)}{1 - F_2(t)}$$

produce un gen de *rată a punerilor în functiune* a sistemului, similară întrucâtva cu rata de defectare definită pentru intervalele în care sistemul este functional. În mod analog se pot evalua media timpului de reînnoire, dispersia acestuia etc.
Un sistem de acest gen poate fi tratat cu metodele de la sistemele cu timp de reînnoire neglijabil dar cu densitatea de repartitie a duratei de viată diferită pe primul interval fată de următoarele. Astfel, un sistem cu reînnoire integrală devine un sistem cu reînnoire generală punând pentru primul interval densitatea de repartitie $f_1(t)$ si pentru următoarele $f_1(t) \otimes f_2(t)$ s.a.m.d.
În domeniul Laplace, pentru momentele repunerii în functiune

$$h_2^*(s) = \frac{f_1^*(s) f_2^*(s)}{1 - f_1^*(s) f_2^*(s)}$$

$$H_2^*(s) = \frac{f_1^*(s) f_2^*(s)}{s[1 - f_1^*(s) f_2^*(s)]}$$

Pentru defectări

$$h_1^*(s) = \frac{f_1^*(s)}{1 - f_1^*(s) f_2^*(s)}$$

$$H_1^*(s) = \frac{f_1^*(s)}{s[1 - f_1^*(s) f_2^*(s)]}$$

Media timpului de functionare se numeste disponibilitate.
Sub aspect organizatoric strategiile de reînnoire elaborate în raport cu caracteristicile de fiabilitate pot fi periodice sau neperiodice. Strategiile tin seama de caracteristicile statistice discutate mai sus.

FIABILITATEA STRUCTURALĂ

Tratarea sistemelor prin observarea stării

Tratarea fiabilitătii unui sistem ca un întreg nu este totdeauna productivă. Deseori se pune problema ca întregul să fie înteles ca o reuniune de subsisteme, fiecare în parte cu caracteristici de fiabilitate proprii.
Este cunoscută reprezentarea sistemelor prin ecuatii de stare si ecuatii de observare care pun în evidentă anumite intrări ale sistemului, anumite variabile de stare si anumite manifestări cantitative observate (iesiri). În general, ecuatiile se prezintă în forma

$$\dot{x}(t) = g[x(t), u(t)]$$
$$y(t) = h[x(t), u(t)]$$

dar este posibilă, în conditiile unei alegeri adecvate a variabilelor de stare $x(t)$, o exprimare mai simplă

$$\dot{x}(t) = g[x(t), u(t)]$$
$$y(t) = h[x(t)]$$

în care observatiile (iesirile) $y(t)$ depind explicit numai de starea sistemului si numai indirect de variabilele de intrare $u(t)$.
Un sistem descompus în părtile lui componente aduce unele din variabilele sale interne, de stare, în calitatea de variabile de interconectare a subsistemelor care îl compun. În contextul nou, unele variabile de stare preiau rolul de intrări (iesiri) ale subsistemelor componente. Ansamblul poate fi descris satisfăcător folosind numai aceste variabile care interconectează diferitele părti ale sistemului.
În studiile de fiabilitate intrările sistemelor sunt considerate solicitările curente, care au un caracter aleator si care nu intră în categoria variabilelor manipulabile. De aceea ele pot fi încadrate în submultimea variabilelor de stare necontrolabile dar al căror efect este observabil. Aceste variabile care fac functionarea sistemului mai mult sau mai putin sigură se grupează într-un vector Γ cu componente dublu indexate, $\gamma_j^{(k)}$, cu indicele inferior care se referă la subsistemul modelat si cu cel superior care numără variabilele pentru acel subsistem. Astfel, subsistemul j poate avea l_j asemenea variabile.
Cu aceste notatii, varabila de iesire de indice i din totalul de p se exprimă astfel:

$$y_i = f(\gamma_1^{(1)}, \gamma_1^{(2)}, \ldots, \gamma_1^{(l_1)}, \gamma_2^{(1)}, \gamma_2^{(2)}, \ldots, \gamma_2^{(l_2)}, \ldots, \gamma_n^{(1)}, \gamma_n^{(2)}, \ldots, \gamma_n^{(l_n)})$$

unde se observă n subsisteme componente, fiecare cu un număr de variabile, care inventariate duc la numărul total

$$N = l_1 + l_1 + \ldots + l_n$$

Sistemul în ansamblul lui se consideră că functionează corect dacă au loc concomitent relatiile

$$y_{i\min} \le y_i \le y_{i\max} \quad (i = 1,2,...,p)$$

în care limitele inferioare si superioare definesc intervale de tolerantă pentru varibilele observate y_i.

Desigur, valorile variabilelor observate sunt aleatoare asa încât fiabilitatea sistemului se poate evalua ca

$$R_S = P[\bigcap_{i=1}^{p}(y_{i\min} \le y_i \le y_{i\max})]$$

în subtext considerându-se că evenimentele intersectate sunt independente, o ipoteză acceptabilă dacă se doreste evitarea unor complicatii de calcul insurmontabile. Valorile y_i pot fi puse în legătură cu variabilele $\gamma_j^{(k)}$, aleatoare la rândul lor

$$\gamma_{j\min}^{(k)} \le \gamma_j^{(k)} \le \gamma_{j\max}^{(k)} \quad (k = 1,2,...,l_j; j = 1,2,...,n)$$

Fiecare componentă are functia proprie de fiabilitate, calculată ca probabilitate a intersectiei unor evenimente independente

$$R_j = P[\bigcap_{k=1}^{l_j}(\gamma_{j\min}^{(k)} \le \gamma_j^{(k)} \le \gamma_{j\max}^{(k)})] \quad (j = 1,2,...,n)$$

De observat că fiabilitatea unei (oricărei) componente este definită în raport cu un criteriu exterior, cel derivat din conditiile de bună functionare a ansamblului. Modelarea fiabilitătii sistemului pe această cale este foarte complicată chiar si pentru sisteme de mică anvergură. Pentru aprecierea fiabilitătii sistemului este necesară evaluarea functiilor de fiabilitate ale fiecărei componente în parte. În context, este presupusă cunoscută toată gama de legi de repartitie ale varabilelor $\gamma_j^{(k)}$. Ulterior, tinând seama de structura sistemului, trebuie calculate variabilele de performantă y_i si densitătile lor de repartitie. Abia după aceea se poate aprecia fiabilitatea sistemului.

O asemenea abordare are o singură sansă: simularea Monte Carlo, dificilă si aceasta din cauza necesitătii de a produce în decursul simulării procesele aleatoare $\gamma_j^{(k)}$.

Tratarea structurală a sistemelor

Analiza fiabilitătii pe baze logic-structurale este mult mai convenabilă si chiar dacă nu este foarte exactă ea este suficient de acoperitoare. Modelele din această categorie se numesc modele logice. În locul variabilelor multiple y_i se utilizează o singură variabilă S bivalentă: $S = 1$ dacă pentru orice $i = 1, 2, ..., p$ avem $y_{i\min} \le y_i \le y_{i\max}$, $S = 0$ dacă pentru un i avem $y_i < y_{i\min}$ sau $y_i > y_{i\max}$. Atunci

$$R_S = P(S = 1)$$

Similar, pentru fiecare subsistem j o variabilă x_j ia valoare 1 sau 0 după cum toti parametrii $\gamma_j^{(k)}$ sunt în limitele permise sau vreunul din ei (cel putin unul) este în afara intervalului îngăduit. Si aici fiabilitatea subsistemului este dată de

$$R_j = P(x_j = 1)$$

Introducerea variabilelor binare S si x_j (j = 1, 2, …, n) crează posibilitatea exprimării primei ca o functie booleană de cele din urmă

$$S = \varphi\ (x_1, x_2, ..., x_n)$$

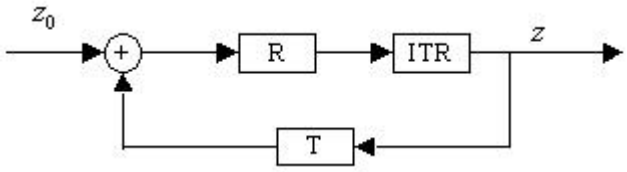
Scopul analizei logico-structurale este stabilirea unei relatii functionale între fiabilitatea sistemului si fiabilitătile subsistemelor componente

$$R_S = \psi\ (R_1, R_2, ..., R_n)$$

Este oare o simplificare în această manieră de modelare? Răspunsul este afirmativ.

Prima simplificare constă în faptul că functiile de fiabilitate ale componentelor sunt cunoscute, deci nu este necesar un calcul prealabil, altminteri destul de complicat, al functiilor de fiabilitate ale fiecărui subsistem. Sunt posibile neconcordante dar calculele sunt considerabil simplificate. A doua simplificare majoră tine de exprimarea booleană mult mai simplă fată de exprimarea functională descrisă mai sus.

Exemplu comparativ. Fie sistemul automat cu reglare după abatere din figură



Modelul functional are în vedere un vector al perfomantelor unidimensional $y = \Delta z/z$. O posibilă descompunere în două subsisteme ar putea fi: un subsistem obtinut prin combinarea regulatorului (R) cu instalatia tehnologică reglată (ITR) si celălalt subsistem traductorul (T) de pe calea de reactie. Functiile de transfer se notează cu $H_d(s)$ pentru primul subsistem si cu $H_r(s)$ pentru conexiunea inversă. Se admite că mărimea de referintă este riguros constantă z_0. Factori aleatori diversi actionează asupra parametrilor din functiile $H_d(s)$ si $H_r(s)$ si îi modifică. În regim stationar functia de transfer conduce la

$$z = \frac{H_d}{1 + H_d H_r} z_0$$

Prin logaritmare si derivare se obtine

$$\frac{dz}{z} = \frac{1}{1 + H_d H_r} \frac{dH_d}{H_d} - \frac{H_d H_r}{1 + H_d H_r} \frac{dH_r}{H_r}$$

care după înlocuirea diferentialelor cu diferente finite exprimă eroarea stationară relativă y în functie de abaterile relative ale amplificării componentelor. Acesta este modelul functional al sistemului. Buna functionare este considerată aceea pentru care $|y| \le \varepsilon$ asa încât fiabilitatea sistemului este

$$R_S = P(|y| \le \varepsilon)$$

Dar

$$|y| = \left|\frac{\Delta z}{z}\right| \le \frac{1}{1 + H_d H_r}\frac{|\Delta H_d|}{H_d} + \frac{H_d H_r}{1 + H_d H_r}\frac{|\Delta H_r|}{H_r} \le \varepsilon$$

admitând că toate amplificările sunt pozitive. Acum, dacă se pune $\varepsilon = \varepsilon_d + \varepsilon_r$, din relatia de mai sus rezultă domeniile de variatie admisibile pentru parametrii sistemului

$$\frac{|\Delta H_d|}{H_d} \le \varepsilon_d (1 + H_d H_r)$$

$$\frac{|\Delta H_r|}{H_r} \le \varepsilon_r \frac{1 + H_d H_r}{H_d H_r} \approx \varepsilon_r$$

Functiile de fiabilitate individuale sunt

$$R_d = P\left[\frac{|\Delta H_d|}{H_d} \le \varepsilon_d (1 + H_d H_r)\right]$$

$$R_r = P\left[\frac{|\Delta H_r|}{H_r} \le \varepsilon_r\right]$$

Pentru a stabili functia de fiabilitate a sistemului pe baza modelului functional trebuie parcurse pentru momente diferite următoarele etape:

1. Stabilirea densităților de repartitie ale parametrilor H_d si H_r;
2. Calculul functiilor de fiabilitate pentru cele două componente;
3. Calculul functiei de repartitie a parametrului de performantă y;
4. Calculul functiei de fiabilitate a sistemului.

Dificultatea majoră este încorporată în etapa a treia.

În varianta logică functionarea corectă este asigurată dacă ambele componente functionează corect

$$S = x_d x_r$$

si functia de fiabilitate a sistemului se exprimă ca

$$R_S = P(S = 1) = P(x_d = 1) P(x_r = 1) = R_d R_r$$

Metode structurale

Modelele structurale pot fi dublate de asa-numitele grafuri de semnal. Un graf de semnal dă ideia de continuitate intrare-iesire pentru o structură conexă complexă. Existenta unui drum de la intrare la iesire este asimilată aici cu functionarea corectă a sistemului.

Modelul *serie* pentru care defectarea fie si a unui singur subsistem scoate sistemul din functiune are graful de semnal din figura alăturată.

Functia care leagă starea de functionare (sau de nefunctionare) a sistemului de starea functională a componentelor este

$$S = x_1 \cap x_2 \cap ... \cap x_n$$

si functia de fiabilitate are expresia

$$R_S = P(S = 1) = \prod_{i=1}^{n} P(x_i = 1) = \prod_{i=1}^{n} R_i$$

În particular, se poate spune că un sistem serie alcătuit din subsisteme fără uzură este la rându-i un sistem fără uzură. Într-adevăr

$$R_S = \prod_{i=1}^{n} e^{-\lambda_i t} = e^{-t\sum_{i=1}^{n} \lambda_i} = e^{-\lambda_S t}$$

Sistemele *paralel* sunt sisteme pentru care defectarea se produce numai în cazul defectării tuturor componentelor. Starea functională a sistemului se leagă de starea componentelor conform relatiei

$$S = x_1 \cup x_2 \cup ... \cup x_n$$

Analiza cantitativă a fiabilitătii sistemului tine seamă de independenta defectărilor. O binecunoscută relatie datorată lui De Morgan permite scrierea

$$\overline{S} = \overline{x_1 \cup x_2 \cup ... \cup x_n} = \overline{x}_1 \cap \overline{x}_2 \cap ... \cap \overline{x}_n$$

cu barele pentru operatia de negare. Stările de functionare si de nefunctionare sunt complementare asa încât se scrie mai întâi, pentru functia de repartitie a duratei de viată

$$F_S = P(\overline{S} = 1) = \prod_{i=1}^{n} P(\overline{x_i} = 1) = \prod_{i=1}^{n} F_i$$

si apoi

$$R_S = 1 - F_S = 1 - \prod_{i=1}^{n} (1 - R_i)$$

Sistemele sunt uzual mai complicate decât cele serie sau paralel. Unele, cele mai putin complexe pot fi combinatii de subsisteme unele serie, altele paralel. Poate fi vorba de o reuniune de intersectii sau o intersectie de reuniuni, deci de secvente de subsisteme legate în graful de semnal în paralel sau de grupe de subsisteme în paralel care la rându-le sunt conectate în serie. Calculul functiei globale de fiabilitate din functiile de fiabilitate individuale nu este deloc complicat în aceste situatii.

Există însă sisteme care nu sunt nici serie, nici paralel, nici serie-paralel si nici paralel-serie. Aceasta se întâmplă când variabila booleană asociată stării de functionalitate a unui subsistem apare în doi sau mai multi termeni (factori) cum ar fi în cazul

$$S = [x_1(x_2 \cup x_3)] \cup [x_3(x_4 \cup x_5)]$$

în care x_3 apare mai mult decât o dată.

În principiu orice functie booleană poate exprima structura unui sistem. Există însă sisteme asa-zis coerente pentru care performantele sunt cu atât mai bune cu cât sunt active (în bună stare de functionare) mai multe subsisteme componente

$$(x_1,x_2,...,x_n)_i \geq (x_1,x_2,...,x_n)_j \Rightarrow (\varphi)_i \geq (\varphi)_j$$

Semnul de inegalitate trebuie înteles ca aplicat *tuturor* componentelor vectorilor binari comparati. De retinut un detaliu: nu toate sistemele reale sunt coerente!

O metodă de tratare generală se bazează pe formula probabilitătii totale. Pentru aceasta variabila (variabilele) care se repetă în functia de structură sunt făcute pe rând 1 si 0. Prin această comutare, functia booleană care leagă starea functională a sistemului de starea componentelor ar putea fi adusă de fiecare dată la una din formele serie-paralel sau paralel-serie. Dacă acesta este cazul, formele acestea permit calculul unor probabilităti conditionate si apoi al functiei de fiabilitate generale, prin evaluarea probabilitătii totale. În etape, se evaluează

$$S/(x_j = 1) = \varphi(x_1,x_2,...,x_{j-1},1,x_{j+1},...,x_n)$$

apoi

$$S/(x_j = 0) = \varphi(x_1,x_2,...,x_{j-1},0,x_{j+1},...,x_n)$$

si în final

$$R_S = P(S = 1) = P(S = 1/x_j = 1)P(x_j = 1) + P(S = 1/x_j = 0)P(x_j = 0) =$$
$$= R_{S/j}R_j + R_{S/\bar{j}}(1 - R_j)$$

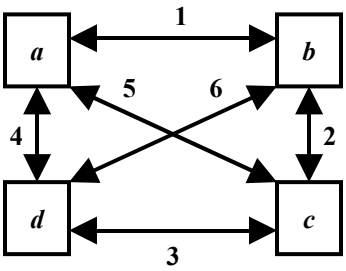
cu notatii evidente. Dacă sistemul cu x_j fixat succesiv la valorile 0 sau 1 nu este combinatie de structuri serie si paralel atunci se aplică metoda probabilitătii totale încă o dată. Dacă structura rezultată prin atribuirea $x_j = 0$ nu este de un tip simplu de tratat, atunci

$$R_{S/\bar{j}} = R_{S/\bar{j}\cap k}R_k + R_{S/\bar{j}\cap \bar{k}}(1 - R_k)$$

Metoda probabilitătii totale permite evaluarea cu usurintă a asa-ziselor ponderi ale fiecărui subsistem în functionarea sistemului definite si notate astfel

$$\frac{\partial R_S}{\partial R_j} = R_{S/j} - R_{S/\bar{j}}$$

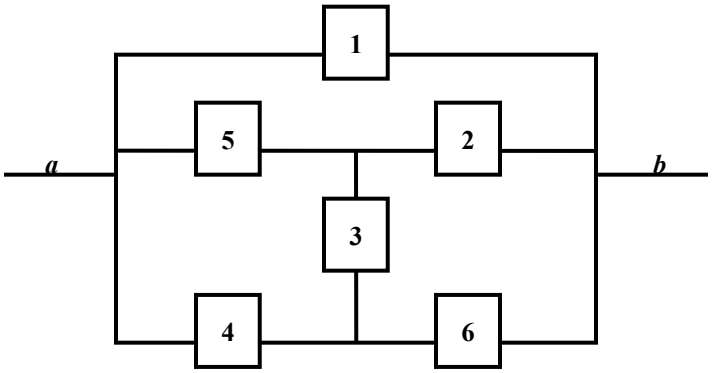
Mai departe este dat un *exemplu*, o retea de comunicatii care conectează patru localităti prin linii directe între fiecare două localităti din sistem, linii permeabile în ambele sensuri.



Dată fiind structura retelei, transmiterea informatiei între două localităti se poate face pe mai multe rute. De pildă, legătura de la (a) la (b) se poate face fie direct, fie pe trasee care includ alte localităti. Functia

$$S = x_1 \cup x_2x_5 \cup x_4x_6 \cup x_2x_3x_4 \cup x_3x_5x_6$$

exprimă posibilitatea ($S = 1$) sau imposibilitatea ($S = 0$) de a conecta cele două localităti. Variabilele binare x_i ($i = 1, 2, \ldots, 6$) exprimă starea de functionare sau nefunctionare a liniilor din figură.
Graful de semnal este reprezentat în figura următoare



Sistemul nu este reductibil la structuri serie si paralel. Variabila x_3, de pildă, se repetă si atunci

$$S / (x_3 = 1) = x_1 \cup x_2x_6 \cup x_4x_5 \cup x_2x_4 \cup x_5x_6 = x_1 \cup [(x_2 \cup x_5)(x_4 \cup x_6)]$$

ceea ce se întâmplă când circuitul (3) este un scurtcircuit si

$$S / (x_3 = 0) = x_1 \cup x_2x_6 \cup x_4x_5$$

pentru o întrerupere pe acelasi circuit. Fiabilitătile conditionate sunt

$$R_{S/3} = 1 - (1 - R)\{1 - [1 - (1 - R)^2]^2\} = -R^5 + 5R^4 - 8R^3 + 4R^2 + R$$

$$R_{S/\bar{3}} = 1 - (1 - R)(1 - R^2)^2 = R^5 - R^4 - 2R^3 + 2R^2 + R$$

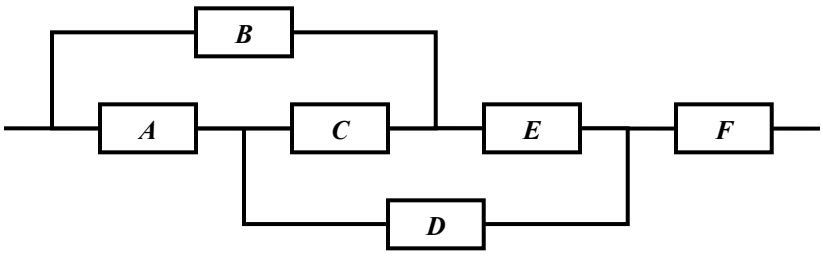
si prin formula probabilitătii totale se obtine fiabilitatea sistemului

$$R_S = R_{S/3}R + R_{S/\bar{3}}(1 - R) = -2R^6 + 7R^5 - 7R^4 + 2R^2 + R$$

Ponderea pentru elementul (3) este

$$\frac{\partial R_S}{\partial R_3} = R_{S/3} - R_{S/\bar{3}} = -2R^5 + 6R^4 - 6R^3 + 2R^2 = 2R^2(1 - R)^3$$

Pentru simplitate, s-a considerat că fiabilitatea oricărei linii la momentul retinut pentru evaluări este R.



Dacă structura sistemului este mai complicată, calculele pot deveni la rândul lor foarte complicate. În asemenea situatii se pot stabili limitele superioară si inferioară pentru fiabilitatea sistemului. O margine superioară este

$$R_{sistem} \leq 1 - \Pi(1 - R_{calea\_i})$$

cu R_{calea_i} fiabilitatea subsistemului alcătuit din modulele serie de pe calea *i*. În produsul din formulă apar factori asociati tuturor căilor paralele. Pentru sistemul exemplificat imediat mai sus: căile care fac sistemul functional sunt în număr de trei, *A*-*D*-*F*, *B*-*E*-*F* si *A*-*C*-*E*-*F* si atunci

$$R_{sistem} \leq 1 - (1 - R_A R_D R_F)(\ 1 - R_B R_E R_F)(1 - R_A R_C R_E R_F)$$

În particular, dacă $R_A = R_B = R_C = R_D = R_E = R_F = R$ atunci

$$R_{sistem} \leq R^3(R^7 - 2R^4 - R^3 + R + 2)$$

Marginea inferioară a fiabilitătii se calculează pe baza asa-numitelor *multimi de tăietură minimă* a diagramei-graf a sistemului. O multime de tăietură minimă este o listă minimală de module constituită astfel încât eliminarea (datorată defectării) a tuturor modulelor din acea listă să facă sistemul disfunct. În cazul în discutie, multimi de tăietură minimală sunt $\{F\}$, $\{A, B\}$, $\{A, E\}$, $\{D, E\}$ si $\{B, C, D\}$.

Marginea inferioară a fiabilitătii sistemului din exemplul dat este partea dreaptă a inegalitătii

$$R_{sistem} \geq \Pi(1 - Q_{taietura\_i})$$

cu $Q_{taietura\_i}$ probabilitatea ca modulele din tăietura minimă *i* să fie toate defecte. În particular, dacă $R_A = R_B = R_C = R_D = R_E = R_F = R$, atunci

$$R_{sistem} \geq R^5(24 - 60R + 62R^2 - 33R^3 + 9R^4 - R^5)$$

FIABILITATEA PROGRAMELOR DE CALCUL

Generalităti

Un program poate fi privit ca o functie care aplică o multime de date care îi sunt furnizate, pe o multime de rezultate. La fiecare executie programul primeste un set de date si poate produce rezultate corecte, într-un fel asteptate, poate produce rezultate eronate sau poate executa operatiuni un timp indefinit, ceea ce echivalează cu a nu produce nici un rezultat. Ultimele două situatii reprezintă defectiuni ale programului. Ele pot fi remediate si programul poate executa din nou calcule până la aparitia unei alte situatii de "pană". Si în cazul studiului fiabilitătii programelor de calcul sunt necesare modele matematice ale comportării modulelor software. De aceea, în continuare sunt prezentate câteva dintre modelele variate pe care literatura le propune.

Modelul Jelinski-Moranda

Sub aspect istoric, modelul Jelinski-Moranda este unul dintre primele modele ale fiabilitătii programelor. Modelul se bazează pe câteva ipoteze. Se admite că:

a) intervalele de timp între defectările succesive sunt variabile aleatoare independente distribuite după legi exponentiale cu parametri posibil diferiti;
b) rata de defectare este proportională cu numărul de erori latente ale programului;
c) la fiecare defectare a programului se efectuează o depanare de durată neglijabilă, prin care se elimină o eroare si numai una.

Conform acestor ipoteze, programul cunoaste un proces de reînnoire cu reînnoiri negative. Rata lui de defectare scade la fiecare defectare/depanare. Între depanări rata defectării este, desigur, constantă deoarece lipseste uzura.

Dacă $N(t)$ este numărul curent de erori rămase (reziduale) atunci $N(0) = N$ este numărul de erori initiale. Cu eliminarea unei erori la fiecare depanare, pentru un interval de functionare X_k, numărul de erori încă prezente pe durata acestui interval este $N(t) = N - k + 1$ cu $t \in [t_{k-1}, t_k]$. Rata de defectare în fiecare interval este constantă si este proportională cu $N(t)$

$$z(t) = \lambda_k = \varphi N(t) = \varphi (N - k + 1)$$

pentru orice interval $[t_{k-1}, t_k]$ de lărgime X_k, $k = 1, 2, ..., N$. Numărul initial de erori N si constanta de proportionalitate φ sunt parametrii modelului.

Functia de fiabilitate în intervalul $[t_{k-1}, t_k]$ este

$$R_k(x) = e^{-\varphi (N-k+1)x}$$

si media duratei între defectările cu numărul de ordine $k - 1$ si k, în ordinea aparitiei lor, este

$$m_k = \frac{1}{\varphi\ (N - k + 1)}$$

Un observator poate face în momentul $t \in [t_{k-1},\ t_k]$ o predictie asupra aparitiei următoarei defectări prin evaluarea functiei de fiabilitate asociate intervalului curent de timp, $R(t,\ t + x)$, si prin calculul duratei medii $m(t)$ a vietii rămase până la următoarea defectare. Aceste functii au expresiile date deja într-un capitol anterior

$$R(t, t + x) = e^{-\int_t^{t+x} z(u)du} = e^{-\varphi (N-k+1)x}$$

si

$$m(t) = \int_0^{\infty} R(t, t + x)dx = \frac{1}{\varphi\ (N - k + 1)}$$

din cauză că rata de defectare în fiecare interval $[t_{k-1},\ t_k]$ este constantă

$$z(t) = \varphi(N - k + 1)$$

si proportională cu numărul erorilor care nu s-au manifestat încă.

Dacă $M(t)$ este numărul de defectări în intervalul $(0,\ t)$ atunci $N = N(t) + M(t)$. Expresia aceasta care descrie în fond un proces de reînnoire permite prognozarea numărului de interventii de efectuat într-un interval oarecare si a numărului de erori reziduale ale programului.

Dacă $P_r(t) = P[M(t) = r]$ reprezintă distributia numărului de defectări în intervalul $(0,\ t)$ cu t fixat, atunci în intervalul initial, până la prima defectare

$$P_0(t) = e^{-\lambda_0 t} = e^{-\varphi Nt}$$

iar conditia initială a procesului este

$$P_r(0) = 0;\ \ r = 1,2,..., N$$

Probabilitatea ca în intervalul scurt $(t,\ t + \Delta t)$ să se producă defectarea k este proportională cu Δt si factorul de proportionalitate este $\lambda_k = \varphi\ (N - k + 1)$.

În intervalul $(t,\ t + \Delta t)$ procesul este descris de sistemul alcătuit din ecuatia cu diferente

$$P_r(t + \Delta t) \approx P_r(t)(1 - \lambda_{r+1}\Delta t) + P_{r-1}(t)\lambda_r \Delta t$$

scrisă pentru $r = 1, 2, ..., N-1$, la care se adaugă ecuatia pentru ultima eroare

$$P_N(t + \Delta t) = P_N(t) + P_{N-1}(t)\lambda_N \Delta t$$

Pentru Δt din ce în ce mai mic sistemul algebric de ecuatii aproximative de mai sus se transformă în sistemul de ecuatii diferentiale

$$\frac{dP_r(t)}{dt} = -\lambda_{r+1}P_r(t) + \lambda_r P_{r-1}(t)$$

$$\frac{dP_N(t)}{dt} = \lambda_N P_{N-1}(t)$$

Cu conditiile initiale mentionate mai devreme, solutia sistemului este

$$P_r(t) = C_N^r (e^{-\varphi t})^{N-r}(1 - e^{-\varphi t})^r$$

pentru orice $r = 1, 2, ..., N$. Factorii $e^{-\varphi t}$ si $1 - e^{-\varphi t}$ sunt probabilităti care se asociază, evident, manifestării unei erori în intervalul $(0, t)$, respectiv eliminării unei erori latente în acelasi interval. Cele două probabilităti complementare intervin într-o lege binomială cu parametrii N si $1 - e^{-\varphi t}$.

Functia de reînnoire, media numărului de defectări în răstimpul $(0, t)$, este

$$H(t) = N(1 - e^{-\varphi t})$$

care are o variatie exponentială. Modelul în discutie are o crestere exponentială a fiabilităti. Densitatea de reînnoire este

$$h(t) = \frac{dH(t)}{dt} = N\varphi\, e^{-\varphi t}$$

Se poate aprecia de asemenea comportarea statistică a numărului de erori remanente la momentul t dat de relatia $N(t) = N - M(t)$, care are în vedere numărul initial de erori si numărul de erori eliminate în urma aparitiei lor în intervalul $(0, t)$.

Din probabilitatea aparitiei a r erori în intervalul $(0, t)$, a cărei expresie este dată mai sus, se poate scrie imediat

$$Q_k(t) = P[N(t) = k] = C_N^k (e^{-\varphi t})^k (1 - e^{-\varphi t})^{N-k}$$

o lege binomială cu parametrii N si $e^{-\varphi t}$. Numărul mediu de erori remanente este $Ne^{-\varphi t}$ si probabilitatea ca toate erorile să fie eliminate în intervalul $(0, t)$ este

$$Q_0(t) = P[N(t) = 0] = (1 - e^{-\varphi t})^N$$

Această relatie permite calculul timpului de testare necesar pentru ca în măsura dată de probabilitatea Q_0 să putem afirma că programul nu mai are nici o eroare

$$t_{Q_0} = \frac{1}{\varphi} \ln \frac{1}{1 - Q_0^{\frac{1}{N}}}$$

De asemenea, se poate evalua timpul mediu necesar eliminării tuturor erorilor

$$D = \frac{1}{\varphi N} + \frac{1}{\varphi (N - 1)} + ... + \frac{1}{\varphi}$$

Dacă se estimează parametrii N si φ din observarea a n erori până la momentul t, atunci se pot face aprecieri importante si interesante asupra comportării programului în continuare. Functia de fiabilitate pentru intervalul $(t, t + x)$ este exponentială cu rata de defectare $\varphi(N - n)$

$$R(t, t + x) = e^{-\varphi (N-n)x}$$

Dacă se impune o anumită probabilitate de bună functionare R, atunci durata x corespunzătoare este

$$x_R = \frac{1}{\varphi (N - n)} \ln \frac{1}{R}$$

si durata medie până la următoarea defectare este

$$m(t) = \frac{1}{\varphi (N - n)}$$

Numărul de defectări în intervalul $(t, t + x)$ se distribuie binomial cu parametrii $N - n$ si φ

$$P[M(t,t+x) = r] = C_{N-n}^{r}(1 - e^{-\varphi x})^{r}(e^{-\varphi x})^{N-n-r}$$

Functia de reînnoire este

$$H(t,t+x) = (N - n)(1 - e^{-\varphi x})$$

Numărul erorilor remanente la momentul $t + x$ este k dacă în intervalul $(t, t + x)$ se produc si sunt remediate $N - n - k$ erori si probabilitatea asociată este

$$Q_k(t+x) = P[N(t+x) = k] = P[M(t,t+x) = N - n - k] =$$

$$= C_{N-n}^{N-n-k}(1 - e^{-\varphi x})^{N-n-k}(e^{-\varphi x})^{k}$$

Durata de testare suplimentară necesară pentru a elimina toate erorile este

$$x_{Q_0} = \frac{1}{\varphi}\ln\frac{1}{1 - Q_0^{\frac{1}{N-n}}}$$

si durata medie până la eliminarea tuturor erorilor este

$$D(t) = \frac{1}{\varphi\ (N - n)} + \frac{1}{\varphi\ (N - n - 1)} + ... + \frac{1}{\varphi}$$

Estimarea parametrilor N si φ se poate face pe baza observatiilor experimentale asupra duratelor succesive de functionare între defectări, $x_1, x_2, ..., x_n$ până la a n-a. Pentru unul dintre intervale, densitatea de repartitie este

$$f_k(x_k / N, \varphi) = \varphi\ (N - k + 1)e^{-\varphi (N-k+1)x_k}$$

Densitatea de repartitie pentru vectorul observatiilor $x_1, x_2, ..., x_n$ este

$$f(x_1, x_2, ..., x_n / N, \varphi) = \varphi^{n}\prod_{k=1}^{n}(N - k + 1)e^{-\sum_{k=1}^{n}\varphi (N-k+1)x_k}$$

Logaritmul acestei functii este o functie de verosimilitate convenabilă pentru maximizat

$$L(x_1, x_2, ..., x_n / N, \varphi) = n\ln\varphi + \sum_{k=1}^{n}\ln(N - k + 1) - \varphi\sum_{k=1}^{n}(N - k + 1)x_k$$

Anularea derivatelor partiale conduce la un sistem de două ecuatii cu necunoscutele N si φ. Valorile rezultate $\hat{N}$ si $\hat{\varphi}$ sunt estimări prin metoda verosimilitătii maxime ale parametrilor teoretici N si φ. Experienta arată că estimatiile au tendinta de a fi infinit, respectiv zero, ceea ce este desigur neconvenabil. În asemenea împrejurări experimentul se prelungeste.

Extinderi ale modelului Jelinski-Moranda

O primă extindere are în vedere rezolvarea la fiecare defectare nu a unei singure erori ci a unei fractii date $1 - c$, constantă, din numărul total de erori N. În aceste conditii, numărul de erori remanente evoluează după schema de mai jos

$$N(t) = \begin{cases} N & t \in [0,t_1) \\ cN & t \in [t_1,t_2) \\ c^2N & t \in [t_2,t_3) \\ \dots & \dots \\ c^kN & t \in [t_k,t_{k+1}) \\ \dots & \dots \end{cases}$$

până la epuizarea tuturor erorilor.
Rata defectărilor se mentine constantă pe intervale si proportională cu numărul erorilor remanente

$$z(t) = \begin{cases} \lambda_0 = \varphi N & t \in [0,t_1) \\ \lambda_1 = \varphi cN = c\lambda_0 & t \in [t_1,t_2) \\ \lambda_2 = \varphi c^2N = c^2\lambda_0 & t \in [t_2,t_3) \\ \dots & \dots \\ \lambda_k = \varphi c^kN = c^k\lambda_0 & t \in [t_k,t_{k+1}) \\ \dots & \dots \end{cases}$$

Parametrii acestei variante a modelului în discutie sunt λ_0 si c cu N si φ neprecizate. Asadar, modelul nu poate prezice nici numărul de defectări într-un interval de timp dat si nici numărul de erori remanente la un moment dat. Se pot însă evalua functia de fiabilitate, durata medie reziduală de viată, se poate prevedea când urmează a se produce următoarea defectare. Astfel

$$R(t,t+x) = e^{-c^k\lambda_0 x} \qquad t_k \le t < t+x < t_{k+1}$$

$$m(t) = \frac{1}{c^k\lambda_0}$$

Metoda verosimilitătii maxime de estimare a celor doi parametri pe baza observatiilor experimentale $x_1,x_2,...,x_n$ – duratele de functionare între defectări până la defectarea a *n*-a – are în vedere densitatea de repartitie a vectorului observatiilor

$$f(x_1,x_2,...,x_n / c,\lambda_0) = \lambda_0^n(\prod_{k=1}^{n} c^{k-1})e^{-\lambda_0\sum_{k=1}^{n} c^{k-1}x_k}$$

care logaritmată produce

$$L(x_1,x_2,...,x_n / c,\lambda_0) = n\ln\lambda_0 + \ln c\sum_{k=1}^{n}(k-1) - \lambda_0\sum_{k=1}^{n} c^{k-1}x_k$$

Prin minimizarea acestei functii se obtin estimatiile $\hat{\lambda}_0$ si $\hat{c}$. Valorile care asigură minimul se obtin prin rezolvarea sistemului

$$\frac{\partial L}{\partial \lambda_0} = 0, \quad \frac{\partial L}{\partial c} = 0$$

în necunoscutele λ_0 si c.

Varianta aceasta a modelului Jelinski-Moranda este cunoscută si sub denumirea de varianta geometrică.
Există si o variantă hibridă care are în vedere două categorii de erori latente: o primă categorie conformă modelului geometric, eliminate în schema geometrică fractionar-constantă; o a doua categorie de erori cu incidentă poissoniană de parametru λ (repartitia Poisson este repartitia pentru variabila aleatoare k discretă, $P(k;\lambda) = \frac{\lambda^k}{k!}e^{-\lambda}$, cu $\lambda > 0$), care permit eventual reluarea programului fără remediere. Modelul Jelinski-Moranda hibrid este cu rată a defectărilor constantă între două defectări succesive

$$z(t) = \lambda_k = \lambda + c^{k-1}\lambda_0 \qquad t \in [t_{k-1}, t_k)$$

Dezvoltarea predictiilor este în bună măsură similară celor expuse mai sus.

Modelele Goel-Okumoto (I) si Musa

Modelul *Goel-Okumoto* (versiunea I) compensează una din rigiditătile modelului Jelinski-Moranda si a variantelor lui. Este vorba de ipoteza rezolvării obligatorii a unei (unor) erori la fiecare defectare. Acest model admite continuarea executării programului fără remediere, remedierea însăsi fiind un eveniment care se poate produce cu o probabilitate precizată p.
Procesul aleator al reînnoirilor nu mai coincide în acest caz cu acela al eliminării erorilor. Dacă $P_r(t) = P[M(t) = r]$ descrie procesul aleator al eliminării erorilor, adică este probabilitatea eliminării a r erori în intervalul $(0, t)$, atunci probabilitatea eliminării celei de a k erori în intervalul $(t, t + \Delta t)$ este produsul dintre probabilitatea ca eroarea k să se manifeste în intervalul specificat, $\lambda_k \Delta t$, cu $\lambda_k = \varphi(N - k + 1)$, si probabilitatea p ca acea eroare să fie eliminată cu ocazia aparitiei ei.
Pentru $r = 0$ se poate scrie ecuatia cu diferente

$$P_0(t + \Delta t) = P_0(t)[1 - p\varphi N\Delta t]$$

care prin trecere la limită se transformă în ecuatia diferentială

$$\frac{dP_0(t)}{dt} = -p\varphi N P_0(t)$$

prin rezolvarea căreia, cu conditia initială $P_0(0) = 1$, se obtine

$$P_0(t) = e^{-p\varphi Nt}$$

Dacă până la momentul t s-au remediat r sau $r - 1$ erori, probabilitatea ca după încă un interval scurt Δt să fie rezolvate (tot) r erori ($r = 1, 2, ..., N - 1$) este aproximativ

$$P_r(t + \Delta t) \approx P_r(t)[1 - p\varphi(N - r)\Delta t] + P_{r-1}(t)p\varphi(N - r + 1)\Delta t$$

De asemenea, probabilitatea ca până la $t + \Delta t$ să se remedieze toate cele N erori este

$$P_N(t + \Delta t) \approx P_N(t) + P_{N-1}(t)p\varphi\Delta t]$$

Ambele relatii cu diferente finite, prin trecere la limită, $\Delta t \to 0$, devin ecuatii diferentiale

$$\frac{dP_r(t)}{dt} = -p\varphi\ (N-r)P_r(t) + p\varphi\ (N-r+1)P_{r-1}(t)$$

$$\frac{dP_N(t)}{dt} = p\varphi\ P_{N-1}(t)$$

Prin rezolvarea sistemului de ecuatii diferentiale rezultat, cu conditiile initiale $P_r(0) = 0$ pentru toti r, se obtine

$$P_r(t) = C_N^r (e^{-p\varphi t})^{N-r} (1 - e^{-p\varphi t})^r$$

o lege binomială de parametri N si $(1 - e^{-p\varphi t})$.

Numărul mediu de erori eliminate este dat de relatia

$$\overline{M(t)} = H(t) = N(1 - e^{-p\varphi t})$$

Repartitia numărului de erori remanente este descrisă de o lege care este tot binomială

$$Q_r(t) = P[N(t) = r] = C_N^r (e^{-p\varphi t})^r (1 - e^{-p\varphi t})^{N-r}$$

Modelul *Musa* este bazat pe aceleasi ipoteze ca si cel anterior. Ia însă în considerare timpul de utilizare a unitătii centrale a calculatorului (CPU). Acest timp este multiplicat cu un factor de compresie c, raportul dintre durata echivalentă de operare si durata de testare. Erorile latente care se manifestă în timpul de testare sunt eliminate cu probabilitatea p. În faza de utilizare propriu-zisă nu mai au loc eliminări de erori. Factorul de contractie exprimă proportia în care executiile din faza de operare curentă au fost reduse prin alegerea metodelor de proiectare si testare. De pildă o oră de testare poate echivala cu mai multe ore de utilizare curentă. Se notează cu $M_0 = N/p$ numărul maxim de defectări, necesar pentru eliminarea tuturor erorilor programului. Tinând seama de factorul de compresie, numărul mediu de defectări observate în intervalul (0, t) devine

$$H(t) = M_0(1 - e^{-p\varphi ct})$$

cu t timpul de rulare curentă a programului. Având în vedere relatia dintre N si M_0 si rezultatul $m_0 = 1/\varphi N$ care exprimă durata medie până la prima defectare, rezultă

$$H(t) = M_0(1 - e^{-\frac{ct}{m_0 M_0}})$$

relatie caracteristică modelului Musa.

Modelele Littlewood si Littlewood-Verrall

Modelele prezentate mai devreme au o limitare dată de faptul că φ este un coeficient constant. *Littlewood* propune un model în care fiecare eroare are ponderea proprie φ_i, $i = 1, 2, ..., N$. Cu această completare, rata defectărilor capătă o expresie nouă

$$z(t) = \lambda_{n+1} = \sum_{i=1}^{N-n} \varphi_i \qquad t \in [t_n, t_{n+1})$$

Desigur, ponderile erorilor nu pot fi în totalitate cunoscute. Se foloseste uzual o distributie apriori subiectivă a acestor ponderi

$$f_a(\varphi_i) = \frac{\beta^{\alpha}(\beta\varphi_i)^{\alpha-1}e^{-\beta\varphi_i}}{\Gamma(\alpha)}$$

o lege Γ, aceeasi pentru orice indice $i = 1, 2, ..., N$, ceea ce subliniază faptul că ierarhizarea erorilor nu este posibilă înainte de a se manifesta.

La momentul $t \in [t_n, t_{n+1})$ când n erori s-au manifestat deja, se poate preciza aposteriori repartitia prin intermediul ecuatiei lui Bayes. Pentru aceasta se observă că probabilitatea ca în intervalul $(0, t)$ eroarea i să nu se manifeste este $e^{-\varphi_i t}$. Ecuatia Bayes dă distributia aposteriori a acelei erori

$$f_p(\varphi_i) = \frac{f_a(\varphi_i)e^{-\varphi_i t}}{\int_0^{\infty} f_a(\varphi_i)e^{-\varphi_i t}d\varphi_i}$$

care după înlocuirea densitătii apriori devine

$$f_p(\varphi_i) = \frac{(\beta + t)^{\alpha}[(\beta + t)\varphi_i]^{\alpha-1}e^{-(\beta+t)\varphi_i}}{\Gamma(\alpha)}$$

asadar o repartitie Γ cu parametrii α si $\beta + t$ cu $t \in [t_n, t_{n+1})$.

Varianta *Littlewood-Verrall*, mai veche, nu cuprinde printre parametri numărul total de erori latente N. Predictiile se referă în acest caz la intervalul de timp cu un început arbitrar si cu finalul la eroarea următoare.

Modele cu rată de defectare variabilă

Aceste modele au în vedere rate de defectare de forma

$$z(t) = (N - k + 1)\varphi(t) \qquad t \in [t_{k-1}, t_k)$$

cu $\varphi(t)$ o functie de timp precizată. Coeficientul φ încetează a mai fi constant sau constant pe intervale.

Dacă functia $\varphi(t)$ este liniară atunci

$$z(t_n + x) = (N - n)\varphi x \qquad x \in [0, t_{n+1} - t_n)$$

si este vorba despre modelul *Schick-Wolverton*. Expresia ultimă arată că rata defectării revine la zero după fiecare defectare/remediere a defectului. În cazul liniar

$$R(t_n, t_n + x) = e^{-\int_0^x z(t_n + x)dx} = e^{-\frac{1}{2}(N-n)\varphi x^2}$$

Durata medie a intervalului până la defectarea următoare este

$$m(t_n) = \int_0^{\infty} R(t_n, t_n + x)dx = \sqrt{\frac{\pi}{2(N - n)\varphi}}$$

Cazul general cu $\varphi(t)$ o functie oarecare este cunoscut sub numele de modelul *Shanthikumar*. Pentru acest caz expresia functiei de reînnoire este

$$H(t) = N[1 - a(t)] = N\left[1 - e^{-\int_0^t \varphi(t)dt}\right]$$

si cea a probabilitătii care descrie procesul aleator al defectărilor este

$$P_r(t) = C_N^r [a(t)]^{N-r} [1 - a(t)]^r$$

o lege binomială de parametrii N si $[1 - a(t)]$ cu $a(t)$ integrala care apare în exponentiala din formula precedentă.

Un caz particular de importantă practică este acela în care ponderea $\varphi(t)$ are expresia

$$\varphi(t) = \alpha\, be^{-bt}$$

ceea ce transformă functia $a(t)$ în

$$a(t) = e^{-\alpha(1-e^{-bt})}$$

Acesta este modelul *Goel-Okumoto* (versiunea II). Procesul de manifestare a erorilor este poissonian

$$P[M(t) = r] = \frac{[\alpha(1 - e^{-bt})]^r}{r!} e^{-\alpha(1-e^{-bt})}$$

În subtext, produsul $N\alpha$ este finit dar N si α tind concomitent la infinit, respectiv la zero, ceea ce echivalează cu un număr de erori foarte mare independente una de cealaltă.

Functia de reînnoire are în acest caz o formă exponentială

$$H(t) = \alpha(1 - e^{-bt})$$

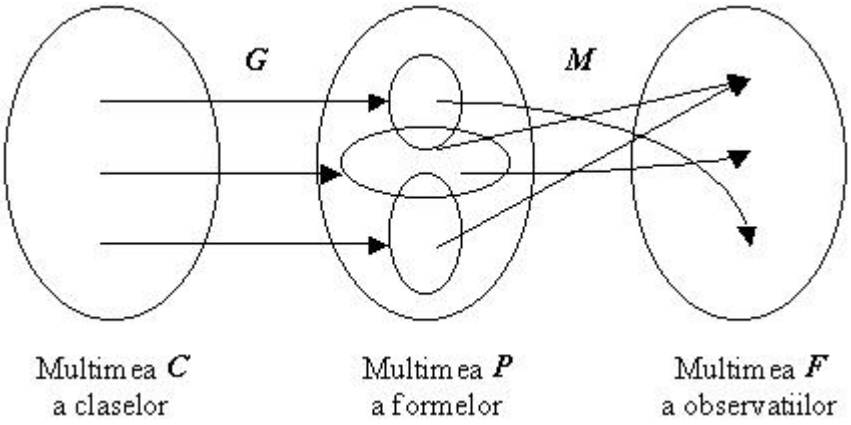
Ca o concluzie a acestei sectiuni se poate retine numărul apreciabil de modele, posibilitătile largi de testare a programelor, dintre care pentru sigurantă se alege uzual situatia cea mai dezavantajoasă.

Varietatea mare de modele oferite de literatură arată cât de importantă este problema functionării sigure a produselor *software*.

DIAGNOZA SISTEMELOR SI RECUNOASTEREA FORMELOR

Generalităti

Diagnoza sistemelor (fault detection) utilizează între altele metoda numită recunoasterea formelor (pattern recognition).
Termenul *forme* asa cum este utilizat în recunoasterea formelor reprezintă o generalizare a ceea ce îndeobste se întelege prin formă când se face referire la geometria unor obiecte, o extindere la formele de manifestare ale unor structuri din natură. În acest cadru general, fenomenele, obiectele si sistemele din natură au anumite forme de manifestare (*patterns*) care fac posibilă distinctia între tipuri/clase diferite de fenomene/obiecte/sisteme.
Într-o exprimare matematică abstractă elementele unui spatiu *C* al claselor sunt asociate prin intermediul unei aplicatii *G* pe un spatiu *P* al formelor (de manifestare). Formele din spatiul *P* sunt asociate la rândul lor prin mijlocirea unei alte aplicatii *M* pe spatiul *F* al observatiilor sau al măsurătorilor.



Numai elementele spatiului *F* sunt nemijlocit accesibile. În general functiile *M* si *G* nu sunt inversabile asa încât trecerea de la spatiul observatiilor înapoi la spatiul formelor si apoi la spatiul claselor pe o cale univocă nu este posibilă.
În circumstantele de mai sus, recunoasterea formelor este o tehnică de a obtine informatia în formă redusă (*reduction*), de a aplica (*mapping*) informatia, de a eticheta informatia (*labeling*).
În procesul de clasificare apare problema dublă a clasificării corecte sau gresite si/sau a posibilitătii de a distinge sau a face confuzie între forme care apartin unor clase diferite.

Recunoasterea formelor prin clasificare, clasificatori

Iată câteva definitii specifice. Prin clasificarea unor forme se întelege asocierea datelor observate uneia sau alteia dintre cele *c* clase prespecificate, pe baza extragerii caracteristicilor/atributelor semnificative si pe baza analizei acestor atribute. Recunoasterea unor forme constă în abilitatea de a clasifica. Uneori se crează o a (*c* + 1)-a clasă care corespunde inclasificabilului (clasa "necunoscut" sau "decizie imposibilă"). O clasă de forme este o multime de forme care împart uzual unele atribute comune, cunoscută fiind originea lor comună. Cheia definirii unor astfel de clase stă în capacitatea de a identifica atribute sau caracteristici potrivite si măsuri adecvate ale similarităţii formelor. Uneori este necesară o preprocesare, o operatie de filtrare sau de transformare a datelor brute pentru a facilita evaluările menite a extrage caracteristici ale formelor si a minimiza zgomotul. Zgomotul este un concept care-si are originea în transmiterea informatiei. În recunoasterea formelor zgomotul reprezintă o serie de adaosuri străine de fenomenul observat cum sunt distorsiunile sau erorile asupra datelor/formelor, erorile în faza de preprocesare, erorile în extragerea caracteristicilor/atributelor, erorile în datele de instruire si de verificare. Un clasificator este uzual o functie dar poate fi si un algoritm care face o partitionare a spatiului caracteristicilor în regiuni de decizie purtând anumite etichete. Dacă vectorul caracteristicilor în particular numerice este *d*-dimensional atunci regiunile sunt o partitie a spatiului R^d. Regiunile sunt prin urmare formate din puncte separate. Exceptie fac *universurile vagi* (*fuzzy*) unde regiunile de decizie se întrepătrund. Între regiunile de decizie există frontiere de decizie. Dacă regiunile sunt definite, atunci procesul de clasificare este simplu. Se aplică unei forme eticheta regiunii căreia îi apartine. Problema definirii acestor regiuni este însă dificilă si este cheia întregii probleme a clasificării. Clasificatorii se bazează pe functii discriminante. Într-o clasificare în *c* clase, functiile discriminant $g_i(x)$, $i = 1, 2, \ldots, c$ actionează după regula *atribuie forma x clasei* w_m (*regiunii* R_m) *dacă* $g_m(x) > g_i(x), \forall\, i = 1,2,...,c; i \neq m$. O frontieră de decizie este definită de egalitatea $g_k(x) = g_l(x),\ k \neq l$.

Instruirea unui sistem de recunoastere a formelor, învătarea formelor de către un astfel de sistem tine seama de experienta sau de cunostintele *apriori* care trebuie totdeauna utilizate la proiectarea unui sistem de recunoastere a formelor. Acele cunostinte se constituie în asa-numitele *multimi de învătare*. Ele constituie o bază de date care furnizează informatii importante asupra modului cum trebuie asociate datele observate cu clase de forme. Instruirea/învătarea utilizează forme tipice, reprezentative pentru formele care apar în aplicatia reală.

Sunt utilizate mai multe variante ale recunoasterii formelor: una este varianta statistică, alta este varianta sintactică/structurală si, mai nou, varianta cu retele neuronale.

Procedurile ingineriei sistemelor de recunoastere a formelor parcurg orientativ următorii pasi:

1. Studiul claselor de forme sub aspectul structural si sub aspectul probabilistic. Explorarea posibilitătilor de definire a unor măsuri ale similarităţii/disimilarităţii între clase/în interiorul claselor. Studiul unor aspecte deformante, al unor proprietăţi invariante si al surselor de zgomot.
2. Determinarea accesibilităţii unor caracteristici/măsurători specifice.
3. Evaluarea performantelor sistemului de recunoastere a formelor raportată la resursele disponibile, acuratetea clasificărilor raportată la resursele *hard*.
4. Disponibilitatea unor date de verificare/instruire (training sets).
5. Disponilbiltatea unor tehnici *de-a gata* de recunoastere a formelor.
6. Dezvoltarea unor posibilităţi de simulare a sistemului de recunoastere a formelor.
7. Verificarea/instruirea sistemului (*training*).
8. Verificarea performantelor sistemului prin simulare.
9. Parcurgerea iterativă a pasilor de mai sus pentru ameliorarea performantelor sistemului de recunoastere a formelor.

În sistemele de recunoastere a formelor se folosesc variate masuri de similitudine. În spatiile metrice, distanta euclidiană

$$d(x,y) = \|x - y\| = \sqrt{(x - y)^T (x - y)} = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$$

sau metrica mai generală

$$d_p(x,y) = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{1/p}$$

sunt utilizate foarte frecvent. Foarte uzuală este si distanta ponderată

$$d_Q^2(x,y) = (x - y)^T Q(x - y) = \|x - y\|_Q^2$$

cu Q o matrice de ponderi pozitiv definită, care dacă este si simetrică se poate factoriza sub forma $Q = T^T T$ si atunci matricea T reprezintă o posibilă transformare de spatiu liniar

$$x_1 = Tx$$
$$y_1 = Ty$$

cu norma euclidiană în spatiul adresă egală cu cea ponderată în spatiul sursă.

Pentru distantele mentionate, care sunt derivate din produsul scalar de vectori $< x, y >$, sunt valabile inegalitatea lui Schwartz si inegalitatea triunghiului. Dacă $x_1 = x / \|x\|$ atunci $< x_1, y >$ este proiectia vectorului y pe directia x.

Dacă vectorii x si y sunt binari de aceeasi lungime atunci este de utilizat distanta Hamming definită ca suma în multimea numerelor naturale a rezultatelor însumării *modulo* 2 a bitilor de acelasi rang ai celor doi vectori.

Pentru multimi finite se foloseste metrica Tanimoto

$$d(A,B) = 1 - \frac{card(A \cap B)}{card(A \cup B)} = 1 - \frac{card(A \cap B)}{card(A) + card(B) - card(A \cap B)}$$

mai ales atunci când elementele multimilor sunt egale ca importantă. În multe situatii, distanta Levenshtien

$$d_L(A,B) = \max\{card(A), card(B)\} - card(A \cap B)$$

este mai potrivită.

Pentru siruri/secvente de numere, fie acestea *u* si *v*, se au în vedere lungimile care pot fi diferite si ordinea elementelor. Elemente utilizate în constructia măsurilor de similitudine si/sau lipsei de similitudine sunt incluziunea (un sir contine un alt sir), suprapunerea (subsirul cel mai cuprinzător comun celor două siruri), similaritatea variațională (costul minim al convertirii unui sir la altul) etc.

Din descompunerea distantei

$$d = \sum (x - y)^2 = \sum x^2 - 2\sum xy + \sum y^2$$

rezultă o contributie constantă irelevantă dată de termenii prim si ultim din expresia desfăsurată si o contributie care poate fi o măsură a similitudinii dată de termenul central. Un maxim al acestuia din urmă produce un minim al distantei între formele *x* si *y*. Termenul central, fără coeficientul –2, este covariatia nenormalizată a celor două caracteristici complete *x* si *y*. Împărtirea cu produsul normelor, dacă astfel de norme sunt definite, conduce la covariatia normalizată sau corelatia caracterisiticilor. Un maxim al acesteia presupune o asemănare/similitudine pronuntată a celor două forme.

Varianta unui spatiu scalat este exemplificată prin forma prototip (*template*) (1 2 3 4) de regăsit în secventa de intrare (7 6 3 4 1 2 4 3 1 2 3 4 5 6 5 4). Prin medierea numerelor două câte două se obtin forma prototip (1,5 3,5) si respectiv, secventa (6,5 3,5 1,5 3,5 1,5 3,5 5,5 4,5) si, după o nouă mediere, în aceeasi manieră se obtin (2,5) si (5 2,5 2,5 5). Recunoasterea se produce în trepte. Este aici de observat economia de calcule multiplicative fată de metoda corelatiei.

Metoda spatiului scalat este generalizabilă prin crearea unei familii de caracteristici

$$\Phi(x,y) = \int_{-\infty}^{\infty} f(x)g(x-u,y)du$$

cu $f(x)$ forma de recunoscut si $g(x, y)$ un nucleu al unor convolutii în care apare si parametrul de scalare *y*.

O functie nucleu foarte utilizată este functia Gauss

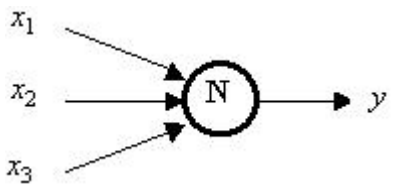
$$g(x,y) = \frac{1}{\sqrt{2\pi}\,y}\exp\left[-\frac{1}{2}\left(\frac{x}{y}\right)^2\right]$$

care este un nucleu de continut unitar adică integrala lui pe axa reală este egală cu 1. El realizează o netezire variabilă cu rezolutia dată de parametrul *y*.

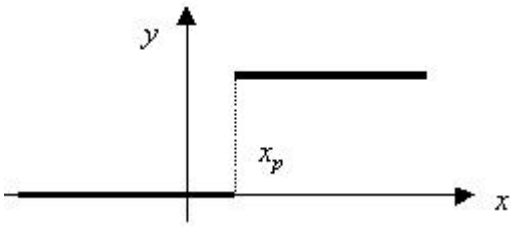
Diagnoză prin retele neuronale artificiale

Amprentele defectiunilor diverse se pot recunoaste prin mijlocirea retelelor neuronale artificiale. Retelele neuronale artificiale sunt reproduceri încă modeste ale retelelor de neuroni ale fiintelor vii, în particular ale celor umane.

Retelele naturale de neuroni sunt cele mai rafinate sisteme de prelucrare a informatiei. Chiar dacă vitezele sunt de cele mai multe ori inferioare celor realizate de calculatoare, o retea cum este creierul uman depăseste în rafinament orice calculator electronic. În tratarea informatiei suntem capabili a percepe, a prelucra semnalele primite, a extrage catacteristici reprezentative dintr-o lume foarte complexă si a decide. Aceste operatii le efectuăm curent, cu o viteză cel putin acceptabilă, în conditiile unei adaptabilităti comportamentale remarcabile vis-à-vis de situatii noi. Această din urmă caracteristică este datorată reflexelor rapide (de pildă dimesionarea pupilară în raport cu intensitatea sursei de lumină) si capacităţii de a învăta. Dacă reflexele sunt în mare măsură similare unor scheme automate simple, capacitatea de a învăta se referă la adaptarea lentă la a executa o actiune nouă (mersul pe bicicletă, de pildă) sau la a aplica o teorie matematică nouă. Un sportiv de performantă repetă de nenumărate ori aceleasi scheme la antrenament până când anumite miscări devin aproape inconstiente. Dobândeste astfel reflexe noi prin învătare. Matematicianul aplică anumite elemente teoretice la rezolvarea unor probleme si prin exercitiu repetat învată să rezolve si să formalizeze aspecte noi ale disciplinei sale.
Neuronul ca celulă de bază a retelelor neuronale are un număr de intrări si o iesire unică. Iesirea poate fi intrare pentru alti neuroni, uzual după o multiplicare cu un anumit număr. Figura alăturată prezintă schematic un neuron cu trei intrări.

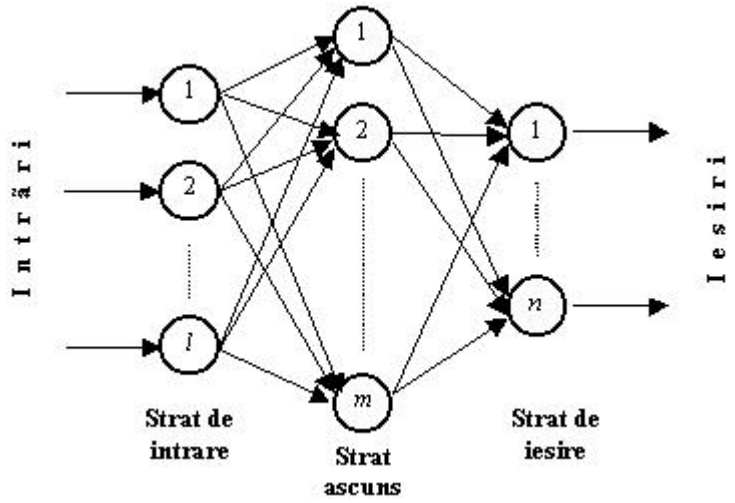


Celula neuronală este caracterizată de o asa-numită functie de activare, care aplică intrările pe multimea valorilor de iesire. Functia de activare pentru celulele neuronale naturale este considerată a fi de forma unui salt marcat de un prag de sensibiltate x_p, conform figurii care urmează.



Variabila x este o combinatie liniară a intrărilor reale multiple, care provin din ambiantă sau de la alti neuroni. Coeficientii acelei combinatii liniare se numesc *ponderi*. Se observă că neuronul are un prag de sensibiltate care produce o

iesire nenulă numai dacă este depăsit. Un x sub pragul x_p face ca iesirea să fie zero (nu produce iesire).
O retea neuronală, fie ea naturală sau artificială este compusă din neuroni interconectati în moduri foarte diverse. Conectarea poate fi ciclică, adică pe o cale mai scurtă sau mai lungă cel putin un neuron din retea îsi serveste siesi intrări, de regulă mijlocit. Asemenea retele se numesc *retele Hopfield* si au parte de o atentie aparte si de o tratare specifică în literatura de specialitate.
Foarte prezente în aplicatiile ingineresti sunt însă *retelele stratificate* pentru care structura este de asa natură încât celulele neuronale sunt organizate în straturi. Se disting un strat de intrare si un strat de iesire, singurele care contin celule în contact nemijlocit cu mediul ambiant. Celulele din stratul de intrare primesc stimuli din exterior, cele din stratul de iesire generează iesiri ale retelei, rezultate ale unor calcule multiple executate predominant în paralel. Mai există unul sau mai multe straturi ascunse formate din celule la care accesul nemijlocit pentru a măsura/observa intrările si/sau iesirile nu este posibil. Conexiunile sunt numai de la un strat la altul într-o ordine a straturilor bine stabilită. Stratul de intrare furnizează intrări primului strat ascuns. Acesta stratului ascuns următor (dacă există un strat ascuns următor). Un penultim strat, si acesta ascuns serveste intrări stratului de iesire. Niciodată nu are loc un transfer de informatie între celulele unui aceluiasi strat de neuroni, niciodată spre un strat anterior. Figura alăturată, care are înfătisarea unui graf orientat cu celule neuronale în noduri si cu legăturile între neuroni pe arce reprezintă tocmai o retea stratificată. Reteaua reprezentată are un singur strat ascuns.



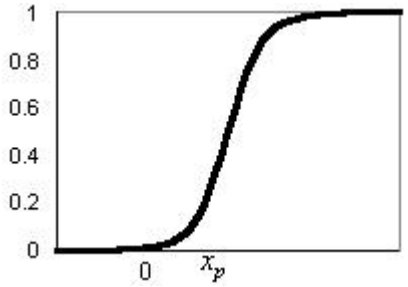
Arcelor li se atasează anumite valori denumite ca si mai devreme *ponderi*. Arcele împreună cu ponderile atasate reprezintă regula matematică de realizare a acelor combinatii liniare a intrărilor simple sau multiple ale fiecărui neuron, intrări provenite din mediul ambiant sau care sunt iesiri ale neuronilor dintr-un strat precedent. Fuctiile de activare dau regula de calcul al iesirilor neuronilor si implicit al intrărilor pentru stratul neuronal următor.

Un element caracteristic al oricărei retele neuronale este capacitatea de învătare. Învătătura acumulată de o retea de neuroni cu functii de activare precizate este stocată în ponderile asociate conexiunilor dintre neuroni. Într-un proces de instruire, cum frecvent se spune în aplicatiile tehnice ale retelelor neuronale artificiale, ponderile sunt aranjate de asa natură încât la intrări similare, răspunsul retelei, cu alte cuvinte iesirile ei să fie similare dacă nu identice. Instruirea unei retele se face pe o multime de perechi intrări-iesiri observate experimental, numită si multime de învătare, multime fatalmente finită. În cursul învătării/instruirii retelei, ponderile sunt ajustate algoritmic pentru ca un anumit criteriu de penalitate să fie minimizat. Intrările sunt uzual valori observate ale unor mărimi fizice. Iesirile pot fi niste etichete (acestea ar putea fi diagnostice, de pildă) sau alte mărimi care sunt legate functional de intrări nu prin relatii functionale clar formulate si deci tratabile prin calcule aritmetice simple ci într-o manieră mai curând tainică, misterioasă. Rezultă din ultima afirmatie că retelele neuronale pot fi utilizate atât la clasificări cât si la interpolări de functii învăluite în mister cum sunt uneori relatiile între variabile.
Clasificările pot avea în vedere simptome ale unei functionări defectuoase a unui organism viu sau a unui sistem tehnic. În cazul acesta multimea de învătare, o multime de perechi intrări-iesiri se clasifică prin etichetare: fiecare set de intrări se asociază cu un dignostic care are eventual un nume. Reteaua neuronală este instruită ca la iesire să producă indicatorul celui mai probabil diagnostic. Astfel instruită, reteaua poate recunoaste diagnosticele respective chiar dacă, cum se întâmplă deseori, simptomele introduse ca intrări nu reproduc riguros simptome-intrări din multimea de învătare. Indicatorii de diagnostic obtinuti la iesire ar putea fi un vector binar, câte un bit pentru fiecare diagnostic. Desigur, iesirea se poate nuanta în numere în intervalul [0,1] care să dea numai o indicatie a diagnosticului/diagnosticelor cel/cele mai probabile. Decizia corectă trebuie sprijinită mai departe pe alte informatii, pe experienta acumulată de experti sau de un sistem expert ca element de inteligentă artificială orientat pe diagnoză.
În procesul de instruire/învătare este necesar un asa-numit criteriu de penalitate care trebuie minimizat prin modificarea ponderilor atasate legăturilor dintre neuronii retelei. Cele mai utilizate criterii sunt cele bazate pe distante, de pildă cel al celor mai mici pătrate. Iesirile observate experimental în conditii de intrări cunoscute, si acestea observate, se constituie în valori tintă pentru învătare. Valorile calculate cu reteaua neuronală trebuie să vină în procesul de învătare cât mai aproape de valorile tintă. Sub aspectul calculului efectiv problema este de a stabili un extrem. Există metode variate de stabilire a extremelor functiilor. Metodele de gradient întâmpină o dificultate majoră în cazul fuctiilor de activare de tipul salt/prag mentionate mai devreme: aceste functii nu sunt derivabile. De aceea functiile de activare pentru retelele neuronale au fost făcute continue si derivabile printr-o usoară modificare. Modificarea conduce la functia sigmoidală care are expresia

$$\sigma(x) = \frac{1}{1 + e^{-\alpha (x - x_p)}}$$

si graficul din figura de mai jos.

Functia sigmoidală este tot de tipul salt dar saltul este neted. Saltul se apropie oricât de mult de saltul net din cazul functiei prag pe măsură ce constanta pozitivă α creste. Functia de activare rămâne însă derivabilă ceea ce este foarte important pentru metodele de gradient.



Referitor la structura retelelor neuronale se pune întrebarea (dublă) naturală: câte straturi de neuroni sunt necesare, câte celule sunt necesare în fiecare strat de neuroni?

Stratul prim, cel de intrare trebuie să conțină atâtea celule câte componente are vectorul intrărilor retelei. Stratul ultim care produce iesirile retelei trebuie să conțină nici mai mult nici mai putin decât numărul de componente ale vectorului de iesire. Rolul oricărui strat neuronal interior/ascuns este acela de a re-formula/re-aplica iesirile stratului anterior pentru a obtine o reprezentare mai clar separabilă, mai limpede clasificabilă a datelor. Straturile interioare sunt cele care permit atasarea unei semantici combinatiilor de intrări ale stratului.

Kolmogorov a dat de timpuriu un răspuns (partial) la problema numărului de celule dintr-un strat ascuns. Răspunsul bazat pe teoria aproximării functiilor sună astfel: fiind dată o functie continuă $\phi : I^d \to R^c, \phi(x) = y$, unde $I = [0, 1]$ si în consecintă I^d este cubul unitate d-dimensional, functia ϕ poate fi implementată într-o retea neuronală cu exact trei straturi, cu d unități (celule) în stratul de intrare, cu $(2c + 1)$ neuroni într-un unic strat ascuns si cu c unități în stratul de iesire.

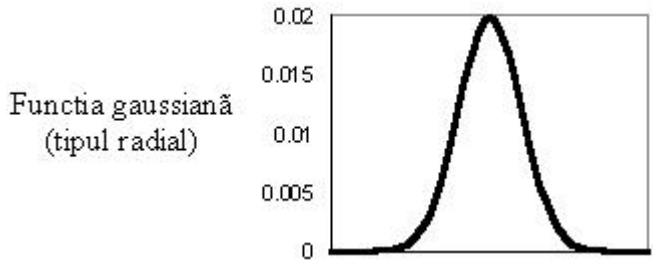
Teorema dată de Kolmogorov este numai o teoremă de existentă. Construirea efectivă a functiilor de activare este deschisă. Posibilitătile de aproximare a functiei ϕ cu functii de un gen sau altul rămâne obiectul unor investigatii de natură mai curând aplicativă.

După cum s-a arătat mai devreme, retelele neuronale artificiale sunt deja larg utilizate pentru a rezolva probleme de învătare în diverse domenii. Prin utilizarea unor date experimentale existente, retelele neuronale "învată" relatiile între intrări si iesiri. Relatiile sunt aproape totdeauna neliniare si sunt cu totul empirice, fără apel la vreo teorie din fundamentele fizicii, ale chimiei etc. Sub acest unghi, retelele neuronale sunt pur si simplu modele regresionale complexe

a căror structură este determinată empiric. Desi retelele neuronale artificiale au fost inspirate încă de la începuturi de retelele de celule nervoase ale organismelor vii, dezvoltările aplicative ulterioare ale acestor retele, până la cele mai recente, cunoscute si sub numele de modele conexioniste sunt produse ale progreselor recente înregistrate de analiza functională.
O retea neuronală tipică (desigur dintre cele stratificate, deocamdată cele mai utilizate) este constituită din mai multe straturi de noduri interconectate, fiecare nod cu o functie de activare si ponderi pe fiecare arc care conectează nodurile retelei între ele. Iesirea fiecărui nod este o functie neliniară de toate intrările sale. Astfel, reteaua este o dezvoltare a relatiei neliniare necunoscute între intrările x si iesirile F într-un spatiu generat de asa-numitele functii de activare ale nodurilor retelei. În particular, învătarea prin propagare directă în retele stratificate poate fi privită ca sintetizarea unei aproximări a unei functii multidimensionale în spatiul generat de functiile de activare $\phi_i(x)$, (i = 1, 2, …, m), adică

$$F(x) = \sum_{i=1}^{m} c_i \phi_i(x)$$

Cu date empirice la dispozitie, cu functiile de activare date si cu topologia retelei cunoscută, parametrii c_i, (i = 1, 2, …, m) sunt ajustati astfel încât eroarea aproximării să fie minimă.
Două tipuri de functii de activare sunt utilizate de obicei: functiile globale si functiile locale.
Functiile de activare *globale* sunt active pe un domeniu larg de valori ale intrărilor si asigură o aproximare globală a datelor empirice. Functiile de activare *locale* sunt active numai într-o vecinătate restrânsă a unei valori de intrare. Efectul lor se estompează pentru valori situate departe de centrul de receptivitate al functiei de activare. Cele mai cunoscute functii de activare globale (exemplificate mai devreme) sunt pragul liniar unitar utilizat în celulele numite perceptron si functia sigmoidală utilizată în retelele cu propagare secventială inversă (BPN – *Back Propagation Network*).
Functiile de tipul radial sunt în esentă locale si sunt utilizate în retelele cu baze de functii radiale (RBFN – *Radial Basis Function Network*). Figura care urmează reprezintă o asemenea functie.



În general, o functie radială asociată unui nod este de forma

$$\phi_i(x) = h\left(\|x - x_i\|\right)$$

Functia gaussiană în varianta ei multidimensională

$$\phi_i(x) = \frac{\sqrt{\det W}}{(2\pi)^{\frac{n}{2}}} \exp\left(-\frac{1}{2}(x - x_i)^T W (x - x_i)\right),\ x \in R^n$$

cu W o matrice pozitiv definită este de tipul radial.
În cazul unidimensional ea se scrie ca

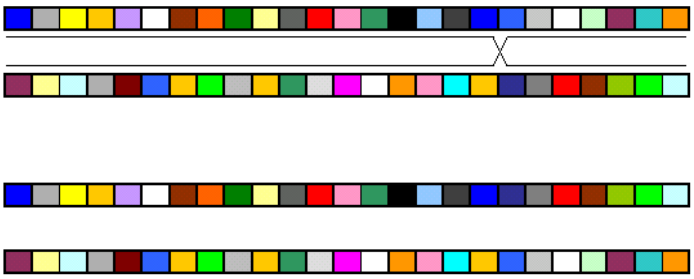
$$\phi_i(x) = \frac{1}{\sqrt{2\pi}\,\sigma_i} \exp\left(-\frac{(x - x_i)^2}{2\sigma_i^2}\right),\ x \in R$$

Retelele de tipul RBFN pot, de asemenea, să aproximeze functiile continue cu o eroare oricât de mică.
Pentru problemele cu dimensionalitate foarte extinsă, acesta este cazul instruirii unei rete neuronale unde varabilele de decizie sunt ponderile, se recurge la metode împrumutate de la regnul viu. Sectiunea imediat următoare contine un asemenea recurs.

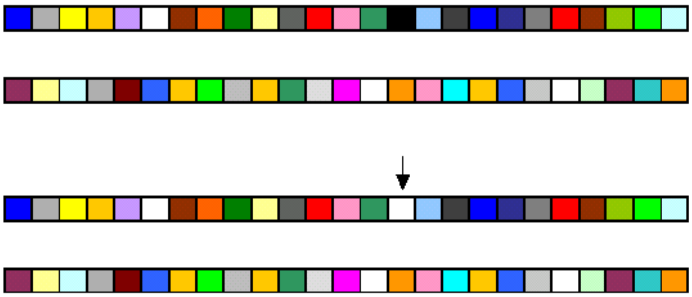
Algoritmi genetici

Problemele ingineresti cu dimensionalitate mare sau foarte mare se pot trata prin metode bazate pe algoritmii genetici. Stabilirea extremelor unor functii multimodale, structurarea optimă si instruirea retelelor neuronale sunt exemple de asemenea probleme. Algoritmii genetici sunt un împrumut din biologie si se bazează pe evolutionismul darwinian.
Se consideră o populatie alcătuită din indivizi descrisi de structuri numite cromozomi. Cromozomii sunt uzual structuri liniare, ansambluri de gene. Figura alăturată ilustrează doi indivizi prin cromozomii lor, genele fiind reprezentate prin culori.

Orice populatie este în evolutie. Indivizii care o alcătuiesc se combină în perechi pentru a genera urmasi. Procedeul curent este cel al combinării-încrucisării. Prin combinare rezultă descendenti care sunt la rândul lor caracterizati de cromozomi. Cromozomii lor rezultă printr-o lectură încrucisată a cromozomilor parentali, în linii mari conform schemei din figura care urmează. În partea de jos sunt reprezentati prin cromozomii specifici descendentii rezultati.
Nu este obligatoriu ca din combinare să rezulte doi descendenti dar în multe aplicatii tehnice aplicarea *operatorului de combinare* produce doi descendenti. Desigur, punctul de comutare a lecturii de la un cromozom la celălalt poate fi pozitionat si altundeva. De asemenea, pot exista si mai multe puncte de traversare.



Lectura cromozomilor parentali se poate face corect dar se poate face si cu eroare. Dacă s-a produs o eroare, se spune că a avut loc o mutatie. Asadar, există un al doilea operator genetic, *operatorul de mutatie*. Figura următoare ilustrează efectul unei mutatii. Sunt prezentati din nou descendentii rezultati prin lectura corectă a genelor, apoi descendentii dintre care unul este afectat de o mutatie la gena marcată cu săgeată.



În aplicatiile ingineresti se vorbeste de populatii de solutii ale unei probleme si de determinarea evolutivă a solutiei acelei probleme. Este vorba mai ales de probleme complexe, de dimensionalitate excesivă pentru care nu sunt căi analitice de solutionare, iar enumerarea tuturor solutiilor acceptabile este o iluzie. Si aici, ca si în cazul populatiilor biologice se vorbeste de adecvarea mai bună sau mai slabă a solutiilor la problema tratată, întocmai cum indivizii unei specii sunt adecvati mai mult sau mai putin la problema supravietuirii într-un mediu generator de variate provocări. Si într-un caz si în altul principiul darwinian al selectiei naturale "*supravietuiesc cei mai adecvati*" lucrează sistematic pentru adaptarea solutiilor la problema fomulată, respectiv a indivizilor la problema supravietuirii si implicit a perpetuării.
Din expunerea generală de mai sus rezultă că problemele tehnice si economice se pot rezolva evolutiv dacă există o codare prin cromozomi adecvati a solutiilor admisibile si dacă se defineste corespunzător o functie de adecvare. Cromozomii din aplicatiile ingineresti pot avea forme diverse. La fel functiile de adecvare. Cea mai frecventă codare este cea binară: cromozomii sunt siruri de biti, genele sunt bitii însisi.
Orice formă ar avea cromozomii, solutionarea unei probleme prin utilizarea algoritmilor genetici parcurge o cale evolutivă, solutia se obtine prin *evolutie*. Algoritmul porneste de la o *populatie* de solutii reprezentate prin *cromozomi*.

Solutiile dintr-o populatie sunt utilizate pentru a forma o nouă populatie de solutii. Motivatia este cât se poate de naturală: speranta că noua populatie va fi mai bună decât populatia veche. Solutiile alese pentru a produce solutii noi, pentru a produce *descendenti*, sunt alese pe baza potrivirii lor cu mediul problemei de solutionat: cu cât sunt mai adecvate, cu atât ele au mai mari sanse de a se *reproduce*.
Procedura este repetată până când s-a generat un număr dat de populatii succesive sau o anumită conditie de adecvare a fost atinsă.
Algoritmii genetici (AG) cuprind în general pasii următori:

1. Generarea aleatoare a unei populatii initiale de *n* solutii acceptabile ale problemei, reprezentate de *n* cromozomi
2. Evaluarea unei functii de adecvare $f(x)$ pentru fiecare cromozom x din populatie
3. Crearea unei populatii noi prin repetarea pasilor următori până ce populatia nouă este completă
    a. Selectia: se selectează o pereche de cromozomi părinti în acord cu adecvarea lor (cu cât sunt mai adecvati cu atât au sanse mai mari de a fi alesi pentru reproducere)
    b. Încrucisarea: cu o probabilitate de încrucisare dată se încrucisează părintii pentru a genera o pereche de descendenti (dacă nu are loc o încrucisare descendentii vor fi cópii identice ale părintilor)
    c. Mutatia: cu o probabilitate precizată se modifică unele pozitii, unele gene din cromozomii descendentilor
4. Populatia generată înlocuieste populatia veche si este folosită pentru o nouă parcugere etapă cu etapă a algortimului
5. Dacă conditia de oprire este atinsă, algoritmul se încheie si se retine solutia cea mai bună din populatia curentă, care este si ultima
6. Dacă conditia de oprire nu este atinsă se reiau evaluările de la pasul **2.**

Liniile generale ale algoritmilor genetici date mai sus au implementări variate.
Una din probleme este, asa cum s-a spus, cum să se creeze cromozomii, cum să se realizeze această codare a indivizilor dintr-o populatie. În functie de forma cromozomilor se definesc cei doi operatori de bază ai algoritmilor genetici, combinarea-încrucisarea si mutatia.
O altă problemă este selectarea judicioasă a părintilor pentru încrucisare. Selectarea se poate face în moduri diferite dar ideea generală este a retine părintii dintre cei mai buni, în speranta că descendentii lor vor fi si mai buni. Poate interveni un dubiu si anume că alcătuirea populatiei noi numai din descendenti ar putea conduce la pierderea cromozomilor cei mai buni din generatia precedentă. Asta se poate întâmpla si, de aceea, se foloseste uneori asa-zisul *elitism*. Asta înseamnă că cel putin una din cele mai bune solutii din generatia curentă este retinută prin copiere în generatia următoare ceea ce o face viabilă poate până în faza finală a evaluărilor.
Modul cel mai obisnuit de codare cromozomică constă în constituirea unei secvente de valori binare.
Cromozomii arată în acest caz astfel:

Cromozomul *k*	1101100100110110
Cromozomul *l*	1101111000011110

Fiecare bit din secventă reprezintă o anumită caracteristică a solutiei. Uneori secventa poate reprezenta unul sau mai multe numere. Desigur, sunt si alte modalităti de codare. Codurile adoptate depind si de tipul problemei de rezolvat. Se pot coda, de pildă, direct numere întregi sau reale, uneori anumite permutări, structuri grafice etc.

Parametri pentru AG. Probabilitătile asociate încrucisării si mutatiei sunt parametri de bază ai algoritmilor genetici. Probabilitătile referitoare la încrucisări se asociază cu frecventa cu care un individ sau altul este selectat în vederea încrucisării: indivizii sau solutiile mai adecvate au probabilităti mai mari de a fi selectati pentru combinare, pentru aplicarea operatorului de încrucisare. Când punctul, altfel aleator, de comutare a lecturii de pe un cromozom pe celălalt este situat chiar pe prima sau pe ultima genă din secventa cromozomială descendentii sunt cópii identice ale părintilor. Încrucisarea este făcută în speranta cât se poate de naturală conform căreia cromozomii noi vor contine genele asociate părtilor bune din cromozomii parentali si acesti noi cromozomi vor reprezenta solutii mai bune ale problemei. Uneori se renunță total la o generatie de solutii de îndată ce o nouă generatie este completă. Alteori este îngăduit ca o parte a populatiei să supravietuiască si în generatia următoare pentru a păstra solutiile cele mai perfectionate ca material genetic valoros pentru încrucisările efectuate în etapa/etapele viitoare.

La mecanismul încrucisărilor se recurge aproape în orice algoritm genetic cu o frecventă mare. Mutatia este folosită mai rar, mai curând ca accident. De aceea probabilitatea de aparitie a unei mutatii este fixată la valori mici, sub 0,1. Mutatia este folosită pentru a preveni stagnarea căutării într-o zonă de adecvare bună numai relativ la o vecinătate restrânsă, ceva analog unui extrem local în optimizare.

Un alt parametru important este dimensiunea populatiei mentinută de regulă constantă de la o generatie la următoarea. Dacă populatia este redusă, diversitatea cromozomială este modestă si algoritmul genetic are posibilităti slabe de încrucisare ceea ce se traduce în conducerea explorării pe un spatiu restrâns. Pe de altă parte populatiile prea numeroase fac ca algoritmii genetici să lucreze lent. O recomandare de luat în considerare are în vedere populatii de zeci de indivizi-solutii.

DIAGNOZĂ PRIN ANALIZA COMPONENTELOR PRINCIPALE

Generalităti

Sistemele automate, în care trebuie incluse si obiectele automatizate sunt supuse posibilitătii de a se defecta si de a devia de la regimul normal de functionare. Efectele functionării defectuoase sunt multiple: reducerea calitătii, diminuarea eficientei, deteriorări partiale ale echipamentelor, opriri ale instalatiilor, producerea unor situatii vecine cu hazardul sau chiar periculoase, impact negativ asupra mediului ambiant etc. Detectarea promptă a defectiunilor si localizarea lor sunt prin urmare de foarte mare importantă.
Problema foarte dezbătută în literatură cunoaste două abordări curente: a) prin metode bazate pe modele; b) prin metode bazate pe cunostinte apriori.
În primul caz este utilizată ipoteza că defectiunile produc modificări ale anumitor parametri fizici care produc la rându-le modificări ale parametrilor si/sau variabilelor de stare ale sistemului. Monitorizarea stării sau parametrilor sistemului face parte din detectarea si diagnosticarea defectiunilor. Dezvoltarea unor modele cuprinzătoare este însă delicată, uneori imposibilă, pentru majoritatea proceselor conduse automat.
În cazul al doilea sunt posibile abordări bazate pe anumite reguli formulate în raport cu structura procesului si din functiile tehnologice ale unitătilor de bază sau pe simulări mai curând calitative. În cazul regulilor, uzual simptomele de rea functionare sunt urmărite în sensul invers al propagării lor. În cazul simulării, modelele calitative descriu comportarea sistemului atât în conditii normale cât si în conditii anormale. Se compară de data aceasta comportarea anticipată pe baza modelului cu observatiile recoltate efectiv de pe sistem. Este asadar necesară crearea în prealabil a unei baze de cunostinte, ceea ce consumă evident timp si efort calificat.
Cum s-a discutat în sectiunea precedentă, o posibilitate relativ putin consumatoare de timp si efort o reprezintă sistemele de diagnoză bazate pe retele neuronale artificiale. Pentru aceasta este necsară o bază de date de instruire care să cuprindă situatii tipice de functionare defectuoasă si simptomele asociate. Prin instruire (*training*) relatia între defectiuni si formele lor de manifestare poate fi învătată sub forma stabilirii ponderilor asociate conexiunilor din retea. Reteaua instruită poate fi apoi utilizată pentru a asocia o functionare defectuoasă ulterioară cu una din situatiile identificate anterior.
Pe durata operării unei instalatii, o serie întreagă de variabile sunt monitorizate si depuse în baze de date. Se spune că de regulă companiile sunt bogate în date dar sunt sărace în informatie, asta mai ales pentru procesele care au principii de functionare vag întelese sau suficient întelese dar foarte greu de modelat. În

aceste cazuri limită, singura sansă si totodată sursă pentru a adânci întelegerea procesului o reprezintă datele produse de măsurători. Analiza mutivariabilă a acestor date este (si) în atentia multor specialisti în probleme de detectare a defectiunilor si în diagnoză. Dat fiind volumul apreciabil de date, acesta trebuie redus la conditia de informatie si metoda utilizată este cunoscută ca *Analiza Componentelor Principale* (ACP) sau ca *Proiectia pe Structura Latentă* (PSL). Ea se bazează pe faptul că practic numai un număr redus de variabile determină procesul si prin ACP/PSL se stabileste un gen de dimensionalitate reală a procesului automatizat. Supravegherea procesului poate fi atunci realizată într-un spatiu al unor variabile latente, spatiu de dimensionalitate rezonabilă. Această tratare este eficientă acolo unde numărul de defectiuni posibile este limitat. Dacă defectiunile sunt mai de numeroase, devine dificil a clasifica si/sau a identifica o gamă mare de functionări neconforme în raport cu posibilitătile de reprezentare oferite de un reper redus ca dimensiuni.

Analiza Componentelor Principale (ACP)

ACP este una din cele mai răspândite metode statistice de analiză multivarabilă. În esentă, variabilele observate afectate de zgomote si mutual corelate sunt reduse la o multime de variabile latente, care sunt un gen de sumar al informatiei relevante, prin proiectarea informatiei brute pe un subspatiu de dimensiune mai redusă. ACP este o procedură de a explica întreaga variatie a datelor observate, grupate într-o matrice $X \in R^{m+n}$, cu m si n numărul de observatii, respectiv numărul de variabile observate. Descompunerea conformă ACP este dată de

$$X = t_1 p_1^T + t_2 p_2^T + ... + t_k p_k^T + E$$

unde t_i si p_i sunt, pentru orice $i = 1, 2, ..., k$, vectori *score* si *loading*, iar E este o matrice de diferente reziduale. Vectorii t_i sunt ortogonali, ca si vectorii p_i la rândul lor, dar cei din urmă sunt în plus de lungime egală cu unitatea. Prima componentă principală este aceea care explică cea mai mare parte a variatiilor, $t_1 = Xp_1$. În spatiul n-dimensional al vectorilor p_i , vectorul p_1 este pe directia celei mai mari variabilităti si vectorul t_1 este proiectia fiecărui vector de observatii pe directia dată de p_1.

A doua componentă principală este aceea care este a doua ca importantă, ca magnitudine. Ea este ca si prima o combinatie liniară a vectorilor variabilelor observate si este ortogonală în raport cu prima. Mai departe, componentele sunt ordonate descrescător, în ordinea contributiei lor la variatia generală a datelor observate. Pentru o matrice X de rang n se pot stabili n asemenea componente. Dacă sunt corelatii puternice si zgomot important atunci se retin numai $k < n$ astfel de componente, care sunt suficiente uzual pentru a explica variabilitatea datelor observate. Restul sunt la nivelul zgomotelor si prin eliminarea lor, procedură obisnuită, se obtine un benefic efect de filtrare. Calculul componentelor principale se poate face pe căi diverse. Una din posibilităti

constă în determinarea asa-numitelor *valori singulare* ale matricei X si apoi descompunerea acesteia conform relatiei

$$X = U\Sigma V^T = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_n u_n v_n^T$$

în care valorile singulare sunt ordonate în ordine descrescătoare: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$. Prima componentă principală este $\sigma_1 u_1$, primul vector *loading* este v_1.

Algoritmul alternativ este cel al celor mai mici pătrate partiale, un algoritm iterativ si neliniar. Acest algoritm calculează pe rând fiecare componentă principală. Perechea t_1, p_1 este calculată din matricea X, celelalte din matricile rezidualelor rezultate la fiecare etapă

$$X = t_1 p_1^T + E_1$$

$$E_1 = t_2 p_2^T + E_2$$

si tot asa în continuare. Algoritmul în detaliu contine pasii de mai jos si se aplică pentru fiecare componentă principală prin înlocuirea matricii X în pasii următori cu $E_i, i = 1,2,\ldots,n-1$.

(1) Se selectează un vector x_j din X si se redenumeste t_1, $t_1 = x_j$;

(2) Se calculează $p_1 : p_1 = t_1^T X / t_1^T t_1$;

(3) Se normalizează $p_1 : p_1 = p_1 / \|p_1\|$;

(4) Se calculează $t_1 : t_1 = Xp_1 / p_1^T p_1$;

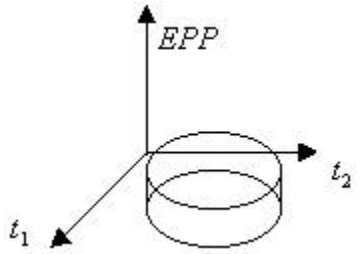
(5) Se compară t_1 utilizat în pasul (2) cu cel calculat la pasul (4); dacă sunt sensibil aceiasi se încheie calculul, dacă nu se reia cu pasul (2).

Detectarea defectiunilor cu ajutorul analizei componentelor principale

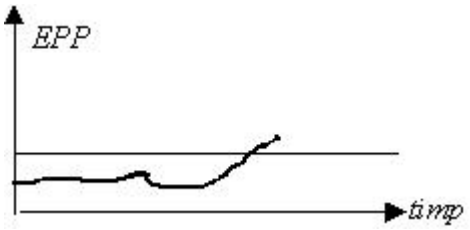
Pentru monitorizarea cu succes a unui proces sunt colectate date si se dezvoltă componentele principale. ACP poate fi sensibilă la scară. De aceea este necesară o prealabilă scalare a datelor, o standardizare a lor. Datele observate se centrează pe valorile medii si se aduc la dipersie unitară (medii si dispersii bazate pe observatiile experimentale). Pe baza acestor date modificate se poate elabora un set de diagrame/nomograme utile în monitorizarea procesului. Pentru fiecare diagramă se poate defini o înfăsurătoare a regimurilor normale de functionare. Odată dezvoltat modelul ACP cu k componente principale, $X = TP^T$, valorile normalizate ale fiecărei variabile x_{ij} pot fi calculate pentru fiecare nouă observatie. Aceste valori pot fi apoi folosite pentru a evalua *eroarea pătratică prezisă*

$$EPP = \sum_{j=1}^{k} (x_{ij} - \hat{x}_{ij})^2$$

În figura următoare axele t_1 si t_2 sunt axele de monitorizare si axa verticală este tocmai *EPP*.



Fiecare nouă observatie este amplasată în cadrul acestei diagrame în noile coordonate t_1, t_2 si *EPP*. Regimurile normale sunt în interiorul cilindrului eliptic (eliptic în cazul zgomotelor gaussiene), regimurile inacceptabile cad în afara acestuia. O monitorizare în timp prin intermediul *EPP* utilizează repartitia χ^2. Figura următoare ilustrează limita de deasupra a cilindrului, dincolo de care apar motive de alertă.



Depăsirea de către *EPP* a limitei de deasupra produce alerta necesară. Proasta functionare a procesului poate produce una din următoarele două situatii: sau defectiunea modifică structural corelatia între variabilele observate si atunci modelul ACP nu mai este valabil deoarece apare posibilitatea unor erori de predictie inadmisibile, sau nu are loc acea modificare a corelatiei si atunci *EPP* rămâne în limite normale dar punctele asociate noilor observatii se plasează înafara înfăsurătoarei regimurilor normale dar sub limita de sus a *EPP*.

Diagnoza prin analiza componentelor principale

Diagnoza presupune specificarea/localizarea defectiunii. Defectiuni diferite produc grupări de puncte (clusters) diferite în spatiul t_i - *EPP* $i = 1,2,...,k$. Diagnosticul poate fi identificat prin inspectarea acestui spatiu, dar în prealabil acestor defectiuni diferite trebuie să li se ia amprentele. Date fiind corelatiile între variabilele observate, aparitia unei defectiuni face ca măsurătorile să se deplaseze într-o directie destul de clar precizată. Prin ACP se poate constitui o bibliotecă de directii asociate cu defectiunile posibile utilizabilă în clasificarea defectiunilor care pot apărea.
Măsurătorile, rezultatele observatiilor sunt grupate în matricea cuprinzătoare $X = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}$, alcătuită din coloanele $x_1, x_2, ..., x_n \in R^{m \times 1}$. Dacă se notează cu $M = \begin{bmatrix} m_1 & m_2 & \dots & m_n \end{bmatrix}$ vectorul mediilor si cu $S = \begin{bmatrix} s_1 & s_2 & \dots & s_n \end{bmatrix}$

vectorul deviatiilor standard calculate din datele nominale ale procesului, $M, S \in R^{1 \times n}$, datele din matricea X pot fi centrate si scalate conform relatiei

$$\overline{X} = \left(X - [1 \quad 1 \quad \ldots \quad 1]^T M\right) diag\left(\frac{1}{s_1} \quad \frac{1}{s_2} \quad \ldots \quad \frac{1}{s_n}\right)$$

Fie acum o defectiune anumită din cele cunoscute, care se si manifestă. Primul vector încărcat asociat defectiunii este centrat si scalat cunform relatiei de mai sus. Prin ACP se obtine o directie în spatiul $t_i, (i = 1,2,...,k)$. Este de observat că vectorii t_i , p_i nu sunt unici. Tot asa de bine si vectorii $- t_i$, $- p_i$ (t_i , p_i cu semnul schimbat) sunt acceptabili sub aspect matematic. Din punct de vedere practic cele două situatii nu sunt echivalente. Lucrurile se pun la punct prin comparatii care recurg la datele experimentale.

Prin punerea laolaltă a mai multor defectiuni se obtine amintita bibliotecă care formal poate fi reprezentată de o matrice $F = [D_1 \quad D_2 \quad \ldots \quad D_N]$ a directiilor (normalizate) D_i ale defectiunilor $i = 1, 2, ..., N$.

Măsurătorile asupra unor variaile de proces care sunt monitorizate curent pot fi analizate prin ACP în modul descris imediat. Fie M_D primul vector (normalizat) încărcat la aparitia unei defectiuni. Alinierea între M_D si directia defectiunii i poate fi măsurată prin produsul scalar $M_D^T D_i$ care este cosinusul unghiului celor doi vectori unitari. Un cosinus apropiat de unitate indică directii paralele sau aproape paralele. Extrema cealaltă, cosinus nul înseamnă ortogonalitate. Cosinusi de valori intermediare reprezintă situatii intermediare. Aceasta este calea de a stabili defectiunea cea mai probabilă: prin compararea unghiurilor făcute de informatia curentă reprezentată de M_D cu directiile asociate defectiunilor din bibliotecă. Se poate marca un prag de diagnosticare τ, subunitar dar apropiat de unitate, fată de care $M_D^T D_i \geq \tau$ să semnaleze prezenta probabilă a defectiunii i.

Diagnoza este deci o problemă de diferentiere a defectiunilor. Este posibilă distinctia clară între diverse defectiuni prin mijlocirea măsurătorilor curente? Distinctia este cu atât mai netă cu cât unghiurile dintre directiile asociate defectiunilor însesi sunt mai mari. Ideal este ca dacă pragul discriminator este τ, atunci

$$D_i^T D_j < \cos(2\cos^{-1}\tau) = 2\tau^2 - 1$$

ceea ce asigură riscuri reduse de clasificare eronată, de confuzii între diagnostice. Cazurile în care conditia nu este îndeplinită pentru toate perechile (D_i, D_j) $i \neq j$ se pot rezolva prin suplimentarea listei de variabile observate, măsurate, prin luarea în considerare a unui al doilea, al treilea s.a.m.d. vector de observatii până la elucidarea situatiei. Prin aceste suplimente de informatie unghiurile dintre directiile defectiunilor din spatiul acesta secundar, redus pot să devină convenabile si riscul erorilor de clasificare poate fi diminuat.

Din alt unghi de vedere, fiind dată biblioteca de defectiuni se poate evalua pragul τ de discriminare

$$\tau > \cos(\frac{1}{2}\cos^{-1}(\max D_i^T D_j)) = \sqrt{(1+\max D_i^T D_j)/2}$$

cu maximul luat pe toate perechile de indici $i \neq j$.

Învătarea de diagnostice noi

Tratarea conform schemei de mai sus poate răspunde si la defectiuni noi, care nu se găsesc în bibliotecă. Defectiunile noi pot fi învătate si depuse în bibliotecă în timpul operării sistemului de diagnosticare. Când sunt detectate anomalii care nu pot fi clasificate în tipurile depuse în bibliotecă, este probabil că o nouă defectiune, necunoscută s-a produs. Directia ei se depune în bibliotecă pentru utilizări viitoare. Astfel biblioteca de diagnostice se actualizează.

DETECTAREA FUNCTIONĂRII NECONFORME SI DIAGNOZA CU FILTRE KALMAN EXTINSE (EKF)

Filtre Kalman extinse

Capitolele anterioare au adus în discutie diverse metode liniare de detectie a defectiunilor si de diagnoză. Partial acele metode sunt susceptibile de a fi aplicate si pentru sisteme neliniare.
Obiectul capitolului prezent îl constituie metodele tipic neliniare bazate pe filtrele Kalman extinse. Pentru problema detectării defectiunilor în functionarea unui sistem cu modele cuplate static si dinamic se foloseste un filtru Kalman care estimează concomitent atât parametrii cât si starea sistemului. În cadrul discret în timp, modelul unui sistem stochastic general poate fi descris matematic de ecuatiile următoare:

$$x_d(t) = f_d[t, x_d(t-1), x_s(t-1); \theta(t-1)] + w_d$$
$$0 = f_s[t, x_d(t), x_s(t); \theta(t)] + w_s$$
$$y(t) = h[t, x_d(t), x_s(t); \theta(t)] + v(t)$$

care sunt, respectiv, modelul dinamic, modelul static si ecuatia de măsurare. Notatiile au semnificatiile următoare:
$v(t)$ – zgomotul aditiv al măsurătorilor;
$w(t)$ – zgomotul aditiv al procesului, la care se adaugă indicii d sau s pentru dinamic sau static;
$x_d(t)$ – variabilele de stare cu dinamică lentă;
$x_s(t)$ – variabilele de stare cu dinamică rapidă;
$y(t)$ – vectorul observatiilor;
$\theta(t) = [c^T(t), d_u^T(t)]$ – vectorul cu coeficientii fizici $c(t)$ si perturbatiile nemăsurate $d_u(t)$.
Pentru a cuprinde variatia temporală a parametrilor se presupune că vectorul $\theta(t)$ variază conform relatiei (*random walk*)

$$\theta(t) = \theta(t-1) + w_\theta$$

Pentru a estima concomitent vectorul de stare si vectorul parametrilor se defineste un vector extins al stării sistemului

$$z(t) = [x_d^T(t)\, x_s^T(t) \theta^T(t)]^T$$

Algoritmul de estimare pe baza secventei de observatii $y(0), y(1), ..., y(t)$ se prezintă astfel:

$$\hat{z}(t) = \bar{z}(t) + K(t)[y(t) - h(t, \bar{z}(t))]$$
$$K(t) = P(t) H^T(t) Q_v^{-1}(t)$$

$$P(t) = M(t) - M(t)H^T(t)[H(t)M(t)H^T(t) + Q_v(t)]^{-1}H(t)M(t)$$
$$M(t) = F(t-1)P(t-1)F^T(t-1) + Q_w(t-1)$$
$$M(0) = M_0$$
$$\bar{z}(0) = z_0$$

cu $H(t) = \frac{\partial h(t,\bar{z}(t))}{\partial z}$, $M(t) = E\{(z(t) - \bar{z}(t))(z(t) - \bar{z}(t))^T\}$ o matrice de covariatie evaluată înainte de măsurătorile de la momentul t, $K(t)$ amplificarea Kalman, $Q_v(t)$ matricea de covariatie a măsurătorilor si $\bar{z}(t)$ estimarea lui $z(t)$ înaintea masurătorii de la momentul t. Matricile $F(t)$ sunt definite prin submatricile lor F_{ij} ($i, j = 1, 2, 3$):

$$F_{11} = \frac{\partial \hat{f}_d}{\partial x_d} \quad F_{12} = \frac{\partial \hat{f}_d}{\partial x_s} \quad F_{13} = \frac{\partial \hat{f}_d}{\partial \theta}$$
$$F_{21} = -\left(\frac{\partial \bar{f}_s}{\partial x_s}\right)^{-1} \frac{\partial \bar{f}_s}{\partial x_d}\frac{\partial \hat{f}_d}{\partial x_d}$$
$$F_{22} = -\left(\frac{\partial \bar{f}_s}{\partial x_s}\right)^{-1} \frac{\partial \bar{f}_s}{\partial x_d}\frac{\partial \hat{f}_d}{\partial x_s}$$
$$F_{23} = -\left(\frac{\partial \bar{f}_s}{\partial x_s}\right)^{-1}\left(\frac{\partial \bar{f}_s}{\partial x_d}\frac{\partial \hat{f}_d}{\partial \theta} + \frac{\partial \bar{f}_s}{\partial \theta}\right)$$
$$F_{31} = 0 \quad F_{32} = 0 \quad F_{33} = I$$

De asemenea, matricea $Q_w(t)$ este definită prin submatricile ei Q_{ij} ($i, j = 1, 2, 3$):

$$Q_{11} = Q_d \quad Q_{12} = -Q_d\left(\frac{\partial \bar{f}_s}{\partial x_d}\right)^T\left(\frac{\partial \bar{f}_s}{\partial x_s}\right)^{-T} \quad Q_{13} = 0$$
$$Q_{21} = -\left(\frac{\partial \bar{f}_s}{\partial x_s}\right)^{-1}\frac{\partial \bar{f}_s}{\partial x_d}Q_d$$
$$Q_{22} = \left(\frac{\partial \bar{f}_s}{\partial x_s}\right)^{-1}\left[Q_s + \left(\frac{\partial \bar{f}_s}{\partial x_d}\right)Q_d\left(\frac{\partial \bar{f}_s}{\partial x_d}\right)^T + \left(\frac{\partial \bar{f}_s}{\partial \theta}\right)Q_\theta\left(\frac{\partial \bar{f}_s}{\partial \theta}\right)^T\right]\left(\frac{\partial \bar{f}_s}{\partial x_s}\right)^{-T}$$
$$Q_{23} = -\left(\frac{\partial \bar{f}_s}{\partial x_s}\right)^{-1}\frac{\partial \bar{f}_s}{\partial \theta}Q_\theta$$
$$Q_{31} = 0 \quad Q_{32} = -Q_\theta\left(\frac{\partial \bar{f}_s}{\partial \theta}\right)^T\left(\frac{\partial \bar{f}_s}{\partial x_s}\right)^{-T} \quad Q_{33} = Q_\theta$$

În expresiile de mai sus

$$\frac{\partial \bar{f}_s}{\partial x} = \left.\frac{\partial f_s}{\partial z}\right|_{z=\bar{z}(t+1)} \quad \frac{\partial \hat{f}_d}{\partial z} = \left.\frac{\partial f_d}{\partial z}\right|_{z=\hat{z}(t)}$$

Matricile Q_d , Q_s si Q_θ sunt matricile de covariatie ale zgomotelor atasate variabilelor dinamice, variabilelor statice, respectiv parametrilor din modelul sistemului.
În general, structura algoritmică a filtrului Kalman extins este aceiasi ca în cazul filtrului Kalman extins pentru modelele dinamice neliniare. Diferenta principală constă în calculul matricii sistemului F si în introducerea matricii generalizate de covariantă a zgomotelor sistemului Q_w care includ termenii interactionali static-dinamic adecvati.

Compensarea filtrelor Kalman extinse

Filtrul Kalman extins are un răspuns adecvat la variatiile parametrice dacă acestia îsi modifică valoarea relativ lent. Contrar acestei cerinte, în sistemele de detectare a erorilor variatia parametrilor necunoscuti poate fi bruscă si poate apărea în orice moment. Întrucât în filtrarea Kalman extinsă matricea de covariatie a erorilor în parametrii necunoscuti scade monoton, filtrul nu poate estima eficient schimbările parametrilor mai târzii în timp. Pentru a preveni o degradare a filtrării de acest gen s-au propus mai multe metode de prevenire a descresterii monotone a amplificarilor filtrului prin anumite conditii suplimentare.
În metoda datorată lui Yoshimura, de pildă, conditia suplimentară care trebuie verificată este

$$\left|\theta_i^n - \hat{\theta}_i(t)\right| > d\left[M_{\theta_i}(t+1)\right]^{1/2}$$

unde θ_i^n si $\hat{\theta}_i(t)$ sunt valorile nominale, respectiv estimatiile parametrilor θ_i, $M_{\theta_i}(t)$ este dispersia erorilor din θ_i, iar d este o constantă pozitivă, uzual 1,96 corespunzătoare unui interval de încredere de 95% pentru o variabilă $z \in N(0;1)$.
Dacă inecuatia de mai sus este verificată pentru cel putin un indice, atunci se modifică $M_{\theta_i}(t)$ si se înlocuieste cu

$$M_{\theta_i}^m(t+1) = \left[\theta_i^n - \hat{\theta}_i(t)\right]^2 \Big/ d^2$$

în plus, valorile θ_i^n se înlocuiesc cu estimatiile lor, $\theta_i^n = \hat{\theta}_i(t)$.

Detectarea schimbărilor datorate fenomenelor nemodelate (defectiunilor)

Când un filtru este actionat de elemente calitative, de cunostinte compilate, el nu modelează obligatoriu corect comportarea sistemului. Este asadar foarte important a stabili dacă filtrul urmăreste comportarea reală a sistemului. Defectiunea sau defectiunile presupuse pot fi atunci eliminate ori acceptate pe baza validitătii modelului filtrant.
Se poate efectua o varietate de teste statistice asupra contributiilor noi (*innovations*) sau asupra rezidualelor pentru a determina cât de valid este

modelul utilizat în proiectarea filtrului. Secventa inovatoare [$\varepsilon(t)$] si matricea ei de covariatie $S_\varepsilon(t) = M[\varepsilon(t)\varepsilon^T(t)]$ sunt date de

$$\varepsilon(t) = y(t) - h(t,\bar{z}(t))$$

$$S_\varepsilon(t) = H(t)M(t)H^T(t) + Q_v(t)$$

Dacă filtrul reflectă corect comportarea curentă a sistemului atunci secventa inovatoare este o secventă aleatoare gaussiană independentă, cu media nulă si covarianta dată de $S_\varepsilon(t)$. Dacă, dimpotrivă, din cauza fenomenelor nemodelate apare o anomalie atunci parametrii statistici ai secventei [$\varepsilon(t)$] se modifică. Pentru supraveghere pe baza unei astfel de secvente de inovare (de reziduale) se poate utiliza testul obtinut prin raportarea secventială a probabilităților.

Pentru fiecare componentă $\varepsilon_i(t)$ se admit ipotezele alternative următoare:

Ipoteza H_0: $\varepsilon_i(t)$ este o secventă aleatoare gaussiană independentă de medie nulă si de dispersie $S_{\varepsilon_i}(t)$;

Ipoteza H_1: $\varepsilon_i(t)$ este o secventă aleatoare gaussiană independentă de medie $a_i(t)$ nenulă si de dispersie $S_{\varepsilon_i}(t)$.

În formularea celor două ipoteze alternative, $S_{\varepsilon_i}(t)$ este componenta din pozitia (i, i) a matricei de covariatie $S_\varepsilon(t)$, iar $a_i(t) = a\sqrt{S_{\varepsilon_i}(t)}$ cu a o constantă adecvată.

Testul raportului secvential al probabilităților este definit ca logaritmul functiei de verosimilitate compusă

$$l_i = \ln \frac{p(\varepsilon_i(1),\varepsilon_i(2),...,\varepsilon_i(t)/H_1)}{p(\varepsilon_i(1),\varepsilon_i(2),...,\varepsilon_i(t)/H_0)}$$

Pentru o secventă aleatoare gaussiană independentă [$\varepsilon_i(t)$] functia din expresia ultimă se poate evalua recursiv cu relatia

$$l_i(t) = l_i(t-1) + a\left[\bar{\varepsilon}_i(t) - \frac{1}{2}a\right]$$

în care $\bar{\varepsilon}_i(t) = \varepsilon_i(t)/\sqrt{S_{\varepsilon_i}(t)}$ este o cantitate normalizată de medie a si cu dispersia 1 (o valoare tipică pentru a este unitatea). Un test al semnului constantei a se realizează prin schimbarea lui a în $-a$. O modificare a testului pentru raportul secvential al probabilităților are forma

$$l_i^*(t) = \begin{cases} l_i(t) \text{ pentru } l_i(t) \geq 0 \\ 0 \text{ pentru } l_i(t) < 0 \end{cases}$$

Cu testul modificat decizia este definită astfel:

- dacă $l_i^*(t) > \lambda_s$ se retine ipoteza H_1
- dacă $0 \leq l_i^*(t) \leq \lambda_s$ se continuă cu o altă observatie.

Pragul λ_s se alege conform unei conditii de timp mediu între alarme false

$$T_{mediu} = \frac{2}{a^2}(e^{\lambda_s} - \lambda_s - 1)$$

sau conform unor probabilităti stabilite pentru alarmă falsă (α) si pentru alarmă ratată (β)

$$(e^{\lambda_s} - \lambda_s - 1) = -\left(B + A\frac{e^B - 1}{1 - e^A}\right)$$

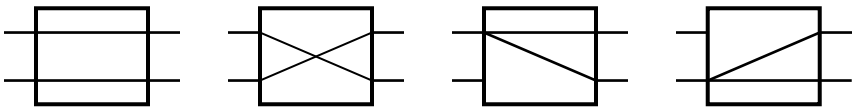
în care $A = \ln[\beta / (1 - \alpha)]$ si $B = \ln[(1 - \beta) / \alpha]$. Pentru $\alpha = \beta = 0{,}05$ rezultă $\lambda_s = 4{,}06$.

FIABILITATEA ÎN RETELE

O particularitate a retelelor este aceea că există aproape tordeauna căi multiple care conectează sursa unui mesaj cu destinatia lui. Totodată există noduri de rezervă care pot fi conectate în retea pentru a înlocui unitătile disfuncte.
Literatura mentionează câteva topologii tolerante la defecte:

- Retelele cu trepte multiple (multi-stage), unele dintre ele suplimentare
- Sitele (meshes) interstitiale
- Crossbarul cu redundante
- Hipercuburile
- Retelele punct-la-punct

Retelele multi-stage lipsite de tolerantă la defcete (retele fluture) sunt alcătuite tipic din comutatoare 2x2, comutatoare cu două intrări si două iesiri.



(a) (b) (c) (d)

Un comutator poate avea una din cele patru setări figurate mai sus:

(a) Directă – linia de intrare superioară conectată la linia superioară de iesire, linia de intrare inferioară conectată la linia inferioară de iesire.
(b) Încrucisată – linia de intrare superioară conectată la linia inferioară de iesire, linia de intrare inferioară conectată la linia superioară de iesire.
(c) Cu difuzare (broadcast) superioară – linia de intrare superioară conectată la ambele linii de iesire.
(d) Cu difuzare (broadcast) inferioară – linia de intrare inferioară conectată la ambele linii de iesire.

Retelele fluture sunt retele în k trepte, cu $k \geq 3$. Au 2^k intrări si 2^k iesiri. Treptele au fiecare cu 2^{k-1} comutatoare. Conexiunile urmează o anumită regulă recursivă de la intrare către iesire.
În stratul de intrare, linia superioară a fiecărui comutator este conectată la intrările unui fluture $2^{k-1}x2^{k-1}$ si linia de iesire inferioară a fiecărui comutator este conectată la intrările unui alt fluture $2^{k-1}x2^{k-1}$. În particular, un fluture cu două straturi, pentru stratul de intrare linia de iesire superioară a fiecărui comutator este conectată la un comutator 2x2 si linia de iesire inferioară este conectată la alt comutator 2x2. Fluturele cel mai simplu, cu un strat, constă dintr-un singur comutator 2x2.

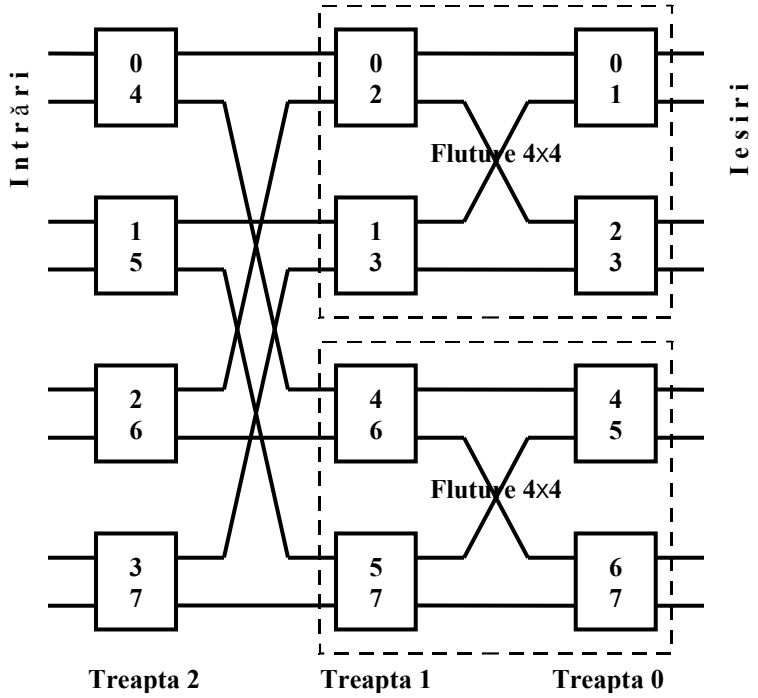
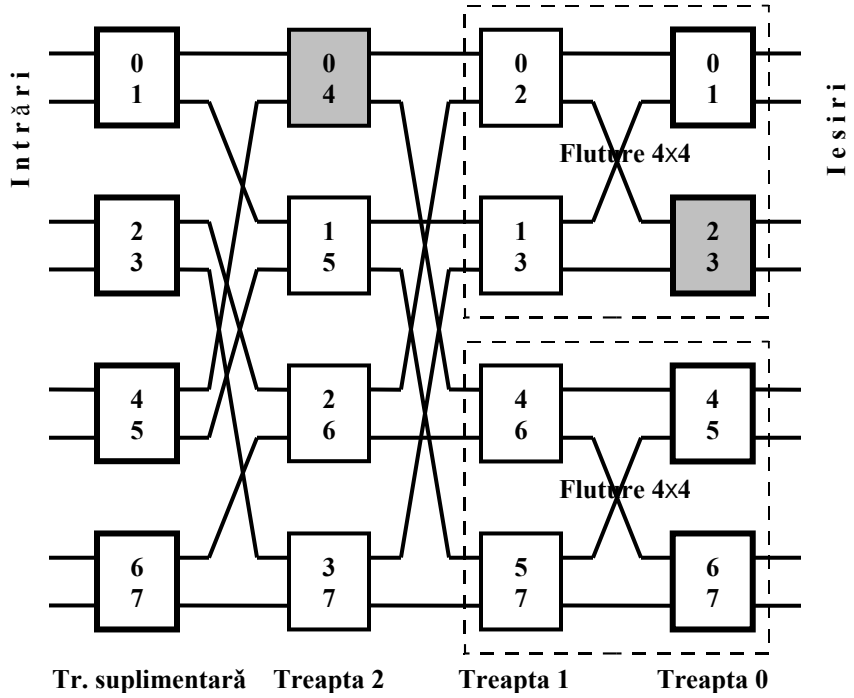
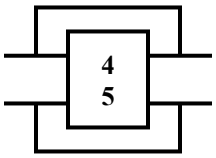


Figura alăturată reprezintă o retea fluture mai complexă, pe care se pot observa unele detalii si se pot preciza unele notatii.
Un comutator din treapta i are liniile numerotate separat cu 2^i. Linia de iesire j a fiecărei trepte merge la linia de intrare j a stratului următor ($j = 0, \ldots, 2^{k-1}$).
Numerele oricărui comutator (switchbox) altul decât din stratul de iesire sunt ambele de aceeasi paritate (pare sau impare).
O retea fluture nu este tolerantă la defecte: există o singură cale de la oricare dintre intrări la o anumită iesire. Dacă un comutator din stratul i clachează, 2^{k-i} intrări sunt deconectate de la 2^{i+1} iesiri.
Sistemul poate încă opera dar într-o manieră degradată.
Pentru a crea o tolerantă la defecte se utilizează retele cu trepte suplimentare.
O posibilitate constă în a adăuga o extra-treaptă prin duplicarea treptei de intrare. Este necesară si o multiplexare în scopul bypassării comutatoarelor din straturile/treaptele de la intrare si de la iesire. Prin adoptarea acestei solutii, un comutator disfunct poate fi ocolit prin rutarea pe bypass.
Exemple:
Comutatorul din stratul 0 cu liniile 2, 3 căzute este duplicat printr-o treaptă suplimentară. Comutatorul disfunct este ocolit (bypassat) prin concursul unui multiplexor.
Comutatorul din stratul 2 cu liniile 0, 4 căzute: extra-stratul este setat astfel ca linia de intrare 0 să fie comutată la linia de iesire 1 si linia de intrare 4 la linia de iesire 5, prin bypassarea cutiei de comutare disfuncte.



Comutatoarele din treapta suplimentară si din treapta ultimă (0), reprezentate cu contur îngrosat, trebuie "citite" fiecare cu posibilitătile ei de ocolire (bypass), ca înfigura imediat următoare.



Se propune ca exercitiu demonstrarea faptului că reteaua cu o treaptă suplimentară poate rămâne conexă în pofida disfunctiei a până la o cutie de comutare oriunde în sistem.

Măsuri ale sigurantei (dependability) unei retele cu mai multe straturi.

Retelele cu interconectare în mai multe trepte conectează N procesoare la N unităti de memorie într-o arhitectură cu memorie partajată ($N = 2^k$). În prezenta elementelor cu defecte, sistemul poate opera, posibil într-un mod degradat.
Rezilienta sistemului în degradare progresivă poate fi măsurată. Iată măsurile uzuale pentru rezilientă:

- Lărgimea de bandă (sau banda de trecere)

- Numărul mediu de căi operationale
- Metrici ale conectivitătii între procesoare si memorii

Toate măsurile sunt functii de timp si presupun că defectele apar si sunt posibil reparate în intervalul [0, t].
Definitiile complete si detaliate ale măsurilor enumerate mai sus sunt:
Banda de trecere $BW(t)$ – numărul mediu (expected) de procesoare la momentul t, care comunică cu (parte din) memorie.
Conectivitatea $Q(t)$ – numărul mediu statistic (expected) de căi procesoare-memorii operationale la momentul t; o cale operațională include un procesor, o memorie si legăturile dintre ele, toate lipsite de defecte.
Un procesor (memorie) este accesibil(ă) (la momentul t) dacă este lipsit(ă) de defecte si este conectat(ă) la cel putin o memorie (un procesor) lipsit(ă) de defecte.
O măsură suplimentară a conectivitătii este cuplul ($A_r(t)$, $A_m(t)$) alcătuit din numărul mediu de procesoare si de memorii accesibile la momentul t.
O altă măsură este dată de perechea ($N_m(t)$, $N_r(t)$) formată din numărul mediu de procesoare (memorii) fără defecte, la care o memorie (un procesor) accesibil este conectat(ă) la timpul t.
Sunt de observat câteva imperfectiuni ale acestor măsuri.
Banda de trecere, $BW(t)$ nu depinde numai de conditiile retelei ci si de volumul solicitării de memorie din partea procesoarelor.
Conectivitatea $Q(t)$ prin numărul de căi nu spune câte procesoare si câte memorii distincte sunt încă accesibile.
Perechea ($A_r(t)$, $A_m(t)$) nu implică faptul că există o retea de interconectare total conexă si fără defecte $A_r(t) \times A_m(t)$; nu indică nici numărul de memorii fără defecte care sunt conectate în medie la un procesor accesibil.
Prin combinarea lui măsurilor $Q(t)$ si ($A_r(t)$, $A_m(t)$) se obtine o caracterizare mai completă a sistemului.
Dacă avem în vedere relatiile

$$N_m(t) = Q(t)/A_r(t) \text{ si } N_r(t) = Q(t)/A_m(t)$$

cuplul ($N_m(t)$, $N_r(t)$) reprezintă o margine superioară pentru sistemul operational mediu maximal deplin conex la momentul t.

Analiza sigurantei (dependability)

O premisă importantă: timpul mediu între căderile (MTBF) componentelor (si posibilele reparatii) este mult mai mare decât durata medie de comunicare între un procesor si o memorie.
O altă premisă importantă: starea componentelor sistemului (cu defecte sau fără) este constantă pentru o perioadă de timp suficient de lungă, atât de lungă ca să permită analiza comportării sistemului într-o stare statistic stationară.
Sistemul este observat la un moment arbitrar t fixat pentru întreaga analiză. Toate măsurile sunt functii de t, inclusiv probabilitătile $p_r(t)$, $p_m(t)$, $p_l(t)$ de bună

functionare pentru procesoare, memorii, legături. De regulă, pentru simplitate, timpul t este omis din notatii (se scrie doar p_r, p_m, p_l).
Se notează cu p_q probabilitatea ca un procesor să aibă nevoie de o legătură cu memoria.

Analiza benzii de trecere

Banda de trecere BW este, cum s-a mai spus, numărul mediu statistic (expected) de procesoare în comunicare activă cu o (parte din) memorie. Se admite o ipoteză simplificatoare: destinatiile cererilor de memorie din partea procesoarelor sunt independente statistic si uniform distribuite pe cele N memorii.
În conditiile specificate, banda de trecere a retelei este produsul numărului de memorii N cu Ψ_m, probabilitatea ca o memorie dată (de pildă memoria 0) să fie lipsită de defecte si să aibă o cerere la intrare ei.
Probabilitătile Ψ_m se calculează iterativ urmând calea care duce la acea memorie desemnată de indicele m. Probabilitatea unei cereri pe o legătură de iesire a unui comutator se calculează din probabilitatea ca o solicitare să fi fost acceptată la legăturile de intrare ale acestui comutator (switch).
Calculul benzii de trecere urmează în linii mari schema explicată în continuare.
O legătură este în starea $X = 1$ ($X = 0$) dacă are (nu are) o solicitare pentru memoria specificată; o legătură defectă este în starea $X = 0$.
Atribuirea de numere celor $k + 1$ straturi ale retelei ($k = \log_2 N$).

- Stratul 0 este ultimul strat; legăturile de iesire sunt conectate la memorii.
- Stratul k este primul si preia iesirile procesorului.

X^i, Y^i sunt iesirile unui comutator din stratul/treapta i.
X^{i+1}, Y^{i+1} sunt intrările unui comutator din stratul i, iesiri a două dintre comutatoarele (diferite) din stratul $i + 1$.

Banda de trecere în cazul retelelor fără redundante

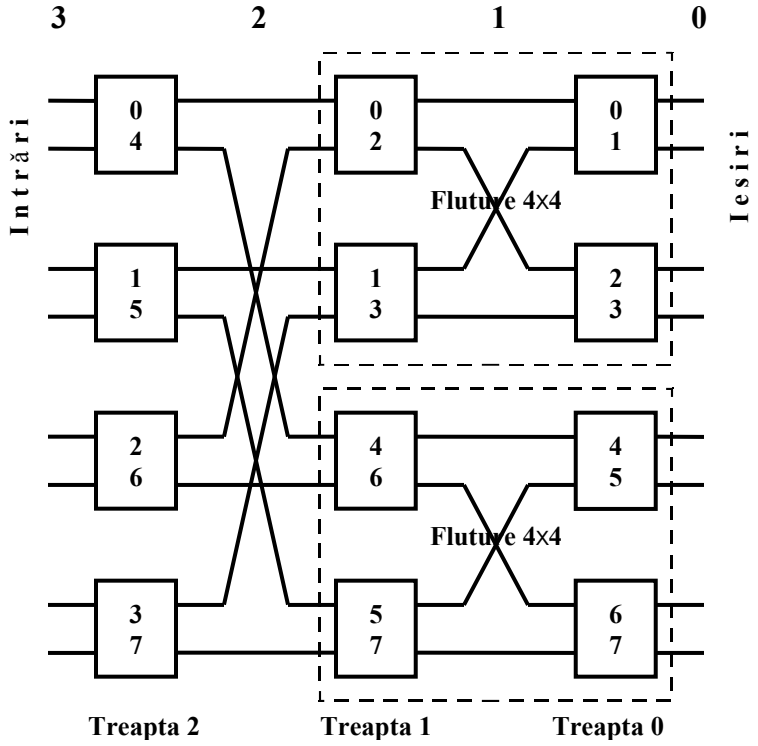
Solicitările de memorie sunt distribuite uniform între memorii; o solicitare care soseste este rutată la oricare legătură de iesire din cele două ale unui comutator, cu probabilitate egală (0,5). În calculul probabilitătii $\Pr(X^i = 1)$, este suficient a lua în considerare numai una din legăturile de iesire.
O cerere către un modul de memorie poate atinge legătura de iesire a unui comutator pe oricare dintre cele două legături de intrare

$$\Pr(X^i = 1) = \Sigma\Sigma_{u,v=0,1}\Pr(X^i = 1/X^{i+1} = u,\ Y^{i+1} = v)\Pr(X^{i+1} = u,\ Y^{i+1} = v) =$$
$$= 0.\Pr(X^{i+1} = 0,\ Y^{i+1} = 0)) + (1/2)p_l\Pr(X^{i+1} = 0,\ Y^{i+1} = 1)) +$$
$$+ (1/2)p_l\Pr(X^{i+1} = 1,\ Y^{i+1} = 0)) + (p_l + (1/4)p_l^2)\Pr(X^{i+1} = 1,\ Y^{i+1} = 1))$$

Sunt luate în considerare numai defectele legăturilor de intrare. Defectele legăturilor de iesire sunt considerate defecte ale legăturilor de intrare ale etapei următoare.
Stările intrărilor unui comutator sunt presupuse a fi independente statistic:

$$\Pr(X^i = u,\ Y^i = v) = \Pr(X^i = u)\ \Pr(Y^i = v)$$
$$\Pr(X^i = 0) = \Pr(Y^i = 0) = 1 - \Pr(X^i = 1)$$

După câteva prelucrări algebrice nu foarte complicate se poate arăta că $\Pr(X^k = 1) = p_q p_r$.



Probabilitatea $\Pr(X^0 = 1)$ se calculează recursiv.

Memoria si legătura ei de intrare pot fi fără defecte: probabilitatea unei astfel de stări este

$$\Psi_m = \Pr(X^0 = 1) p_l p_m$$

si în final

$$BW = N\,\Psi_m$$

Conectivitatea retelelor de interconectare fără redundante

Cum s-a definit, Q este numărul mediu de căi operationale pentru perechi procesor-memorie conectate. În lipsa oricăror redundante există exact o cale între un procesor si o memorie.

În cuvinte, conectivitatea Q este produsul numărului de perechi procesor-memorie cu probabilitatea existentei unei căi fără defecte. Probabilitatea aceasta este $p_r p_l^{k+1} p_m$, cu $k + 1$ numărul de legături pe cale, adică numărul de trepte + 1 ($k = \log_2 N$).

Deoarece numărul de căi procesor-memorie este N^2, rezultă

$$Q = N^2 p_r p_l^{k+1} p_m.$$

Calculul măsurilor aditionale pentru retelele de interconectare fără redundante

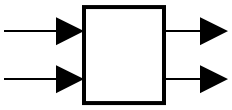
Am notat cu A_r numărul mediu de procesoare accesibile. A_r este produsul numărului de procesoare N cu probabilitatea ϕ_r ca un procesor (de pildă procesorul 0) să fie accesibil.
O legătură este în starea $X = 0$ ($X = 1$) dacă toate (nu toate) căile de la procesor la memorii sunt defecte.

- O cale defectă este o cale cu cel putin o legătură defectă.
- O legătură defectă este în starea $X = 0$.

Numerotarea treptelor se face si de această dată de la k la 0; X^i exprimă starea legăturii în treapta i. În particular, $\Pr(X^0 = 1) = p_m$; $\Pr(X^0 = 0) = 1 - p_m$.
O ecuatie recursivă

$$\Pr(X^{i+1} = 1) = p_l[1 - \Pr(X^i = 0)^2]$$

este suportul evaluărilor curente.



În cele din urmă

$$\phi_r = p_r p_l \Pr(X^k = 1)$$

si

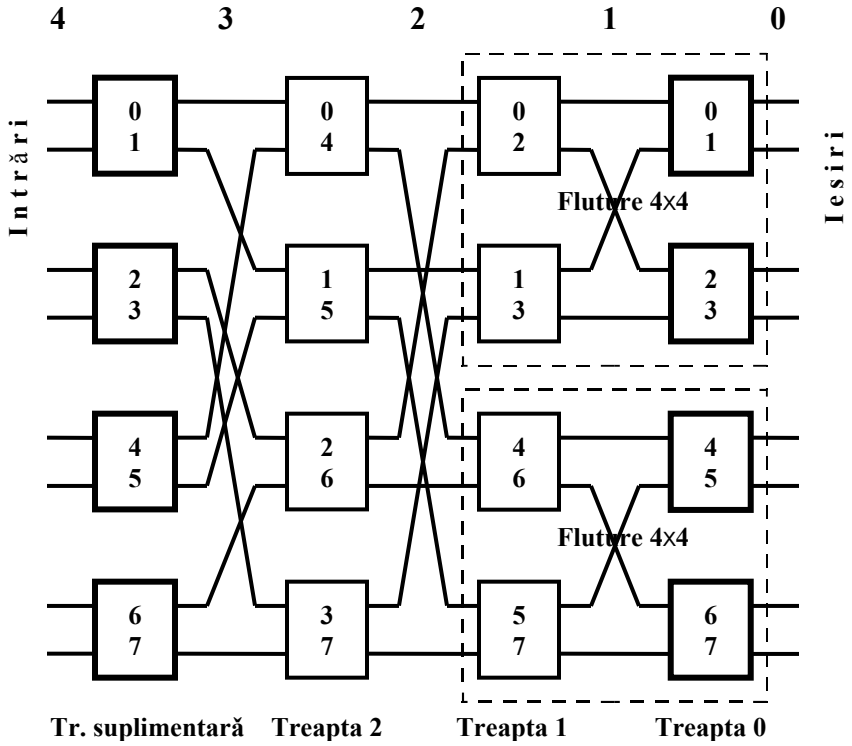
$$A_r = N \phi_r$$

Numărul mediu de memorii accesibile, A_m se obtine pe o cale similară, prin schimbarea probabilitătii p_r cu probabilitatea p_m.
Urmează un *exemplu* numeric.
Calculul benzii de trecere, în ipoteza unor legături fără defecte, în cazul $N = 8$, $k = 3$, pentru un moment t fixat.
$p_r = 0{,}8$; $p_m = 0{,}9$; $p_l = 1$ (legături fără defecte); $p_q = 0{,}7$.
Calculul benzii de trecere:
$\Pr(X^3 = 1) = p_q p_r = 0{,}56$
$\Pr(X^2 = 1) = 0{,}56 - 0{,}25\text{x}0{,}562 = 0{,}536$
$\Pr(X^1 = 1) = 0{,}536 - 0{,}25\text{x}0{,}5362 = 0{,}464$
$\Pr(X^0 = 1) = 0{,}464 - 0{,}25\text{x}0{,}4642 = 0{,}41$
$BW = 0{,}41Np_m = 0{,}41\text{x}8\text{x}0{,}9 = 2{,}95$
Calculul conectivitătii si al măsurilor suplimentare:
$Q = N^2\text{x}0{,}8\text{x}0{,}9 = 0{,}72N^2 = 46{,}08$
$A_r = Np_r[1 - (1 - p_m)^N] = 0{,}8N(1 - 0{,}1^N) \approx 0{,}64$
$A_m = Np_m[1 - (1 - p_r)^N] = 0{,}9N(1 - 0{,}2^N) \approx 0{,}72$

$N_r = Q/A_m = 64$
$N_m = Q/A_r = 72$
Asupra acestor ultime rezultate sugerăm cititorului o discutie.

Retele fluture si retele fluture cu trepte suplimentare

În reteaua fluture fără redundante, cele două intrări în orice comutator sunt considerate independente statistic. Într-o retea cu o treaptă în plus sunt câte două căi care conectează orice pereche procesor-memorie. Legăturile sunt de data aceasta dependente si ecuatiile date mai sus nu mai sunt valide.
Asa cum s-a mai spus, primul si ultimul strat au multiplexoare si demultiplexoare pentru care analiza este diferită de aceea a etajelor interioare.
Sunt patru legături care duc la două comutatoare, două perechi sunt independente, desi legăturile din aceeasi pereche sunt dependente.



Exemplu: legăturile de iesire 0 si 1 din etajul 2 sunt dependente (procesoarele 0 si 1 trimit solicitări către memoria 0 prin ambele); legăturile 2 si 3 sunt si ele dependente; perechile 0, 1 si 2, 3 sunt însă independente.

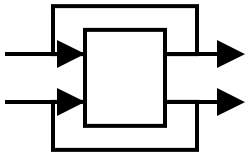
Banda de treere pentru o retea cu trepte suplimetare

Banda de trecere (*BW*), reamintim, este numărul mediu de procesoare care comunică activ cu (o parte din) memorie si este acelasi lucru cu numărul mediu de memorii care comunică activ cu vreun procesor. Banda de trecere se obtine ca produsul numărului de memorii N cu Ψ_m, probabilitatea ca o memorie dată (de pildă memoria 0) să fie fără defect si să aibă o solicitare la intrarea ei.
Ca si altădată, Ψ_m se calculează iterativ, urmând o cale de la procesor care duce la o anumită meorie.
Legătura este în starea 1 (0) dacă ea are (nu are) o solicitare de memorie; o legătură defectă este în starea 0.
Calculul benzii de trecere pentru reteaua din figură urmează pasii prezentati imediat.
Sunt $k + 2$ trepte numerotate $k + 1, k, \ldots, 0$ (cu $k = \log_2 N$). Se notează

- X^i, Y^i – starea celor două legături de iesire din etajul i
- X^{i+1}, Y^{i+1}, Z^{i+1}, W^{i+1} – starea intrărilor pe legăturile din etajul i, totuna cu legăturile de iesire pentru etajul $i + 1$.

Probabilitatea ca o intrare din etajul i să aibă solicitare este calculată pe baza probabilitătii ca o solicitare să fie acceptată la legăturile de intrare.
Pentru primul etaj ($k + 1$ – etajul procesorului) se poate scrie

$$\Pr[X^{k+1} = 1] = p_q p_r;\ \Pr[X^{k+1} = 0] = 1 - \Pr[X^{k+1} = 1]$$



Pentru etajul k (procesoarele sunt independente statistic):

$$\Pr[(X^k, Y^k) = (0, 0)] =$$
$$= (\Pr[X^{k+1} = 0])^2 + q_l^4(\Pr[X^{k+1} = 1])^2 + q_l^3 2\Pr[X^{k+1} = 0]\Pr[X^{k+1} = 1]$$
$$\Pr[(X^k, Y^k) = (0, 1)] =$$
$$= (1 - q_l^3)\Pr[X^{k+1} = 0]\Pr[X^{k+1} = 1] + (1 - q_l^2)q_l^2(\Pr[X^{k+1} = 1])^2$$
$$\Pr[(X^k, Y^k) = (1, 0)] = \Pr[(X^k, Y^k) = (0, 1)]$$
$$\Pr[(X^k, Y^k) = (1, 1)] = (1 - q_l^2)^2(\Pr[X^{k+1} = 1])^2$$

Treptele interne în retelele cu trepte suplimetare

Expresiile de mai devreme au presupus că o solicitare este trimisă mai întâi prin conexiunea directă si se foloseste conexiunea încrucisată numai dacă cea directă este disfunctă.

- Protocol diferit: conexiunile directă si încrucisată cu probabilităti egale, expresiile pentru probabilităti vor fi diferite.

Pentru etajele interioare ($i = k - 1, \ldots, 1$):

$$\Pr[(X^i, Y^i) = (u, v)] =$$

$$= \sum_{(s_0,s_1,s_2,s_3)=(0,0,0,0)}^{(s_0,s_1,s_2,s_3)=(1,1,1,1)} \Pr[(X^{i+1}, Y^{i+1}, Z^{i+1}, W^{i+1}) = (s_0, s_1, s_2, s_3)] \cdot$$

$$\cdot \Pr[(X^i, Y^i) = (u, v) \mid (X^{i+1}, Y^{i+1}, Z^{i+1}, W^{i+1}) = (s_0, s_1, s_2, s_3)] =$$

$$= \sum_{(s_0,s_1,s_2,s_3)=(0,0,0,0)}^{(s_0,s_1,s_2,s_3)=(1,1,1,1)} \Pr[(X^{i+1}, Y^{i+1}) = (s_0, s_1)] \Pr[(Z^{i+1}, W^{i+1}) = (s_2, s_3)] \cdot$$

$$\cdot \Pr[X^i = u \mid (X^{i+1}, Z^{i+1}) = (s_0, s_2)] \Pr[Y^i = v \mid (Y^{i+1}, W^{i+1}) = (s_1, s_3)]$$

pentru $u, v = 0, 1$.

Numai probabilitătile combinate (joint) ale celor două legături sunt necesare. Acestea pot fi calculate recursiv de la etajul $k + 1$ (etajul procesor) la etajul 0 (etajul de memorare).

Etajul 0 include demultiplexoare

$$\Pr[X^0 = 1 | (X^1, Y^1) = (0, 0)] = 0$$

$$\Pr[X^0 = 1 | (X^1, Y^1) = (0, 1)] = (1/2)p_l$$

$$\Pr[X^0 = 1 | (X^1, Y^1) = (1, 0)] = (1/2)(1 - q_l^2)$$

$$\Pr[X^0 = 1 | (X^1, Y^1) = (1, 1)] = (1/2)(3p_l - p_l^2) - (1/4)p_l(1 - q_l^2)$$

$$\Psi_m = \Pr(X^0 = 1)p_l p_m$$

În final:

$$BW = N\Psi_m$$

Conectivitatea pentru o retea cu trepte suplimetare

Q este produsul numărului de perechi procesor-memorie N^2 cu probabilitatea ca între componentele perechii să existe cel putin o cale fără defecte.

Fiecare pereche procesor-memorie este conectată prin două căi disjuncte (sentelege, cu exceptia celor două capete).

Probabilitatea ca cel putin o cale să fie fără defecte este egală cu probabilitatea ca prima cale să fie fără defecte adunată cu probabilitatea ca cealaltă cale să fie fără defecte din care trebuie scăzută probabilitatea ca ambele căi să fie fără defecte.

Probabilitatea poate asuma una din cele două expresii (a se compara calea între procesorul 0 si memoria 0 cu calea între procesorul 0 si memoria 1).

Calculul conectivitătii urmează pasii de mai jos.

Pentru căile dintre procesorul 0 si memoria 0:

$$\Pr(\text{cel putin o cale este fără defecte}) = \Pr(0, 0) =$$

$$= 2p_r(1 - q_l^2)p_l^{k+1}p_m - p_r p_l^{2k+2}(1 - q_l^2)^2 p_m$$

Pentru căile dintre procesorul 0 si memoria 1:

$$\Pr(\text{cel putin o cale este fără defecte}) = \Pr(0, 1) =$$

$$= p_r(1 - q_l^2)p_l^k(1 - q_l^2)p_m + p_r p_l^{k+2}p_m - p_r p_l^{2k+2}(1 - q_l^2)^2 p_m$$

Jumătate din perechile procesor-memorie urmează valoarea Pr(0, 0) si cealaltă jumătate urmează valoarea Pr(0, 1).

$$Q = [\Pr(0, 0) + \Pr(0, 1)]N^2/2$$

Măsurile aditionale pentru o retea cu trepte suplimetare

A_r si A_m sunt numărul mediu de procesoare accesibile, respectiv numărul mediu de memorii accesibile.
ϕ_r (ϕ_m) sunt probabilitătile ca un procesor (o memorie) dat(ă) să fie conectat(ă) la cel putin o memorie (un procesor).
Pentru calcularea lui A_r se face aceeasi descriere de stări: legătura este în starea $X = 0$ ($X = 1$) dacă toate (nu toate) căile de la procesor la memorii sunt defecte.
O cale defectă este o cale care contine cel putin o legătură defectă.
O legătură defectă este în starea $X = 0$.
Numerotarea etajelor se mentine, $k + 1$ (procesoarele) la 0 (memoriile).
Dacă X^i descrie starea legăturii din etajul i

$$\phi_r = p_r p_l \Pr(X^{k+1} = 1)$$

si

$$A_r = N\,\phi_r$$

Urmează calculul măsurii A_r.
$\Pr(X^i = 1)$ se calcuează recursiv de la treapta 0 la treapta $k + 1$.
X^i, Y^i notează si acum starea celor două legături din etajul i.
Pentru etajul 0:

$$\Pr(X^0 = 1) = p_m \text{ si } \Pr(X^0 = 0) = 1 - p_m.$$

Pentru etajul 1:

$$\Pr[(X^1, Y^1) = (0, 0)] = \{\Pr[X^0 = 0]\}^2 + 2\Pr[X^0 = 0]\Pr[X^0 = 1]q_l^3 + \\ + \{\Pr[X^0 = 1]\}^2 q_l^6$$

$$\Pr[(X^1, Y^1) = (0, 1)] = \Pr[X^0 = 0]\Pr[X^0 = 1][q_l(1 - q_l^2) + q_l^2 p_l] + \\ + \{\Pr[X^0 = 1]\}^2 q_l^3(1 - q_l^3)$$

$$\Pr[(X^1, Y^1) = (1, 0)] = \Pr[(X^1, Y^1) = (0, 1)]$$

$$\Pr[(X^1, Y^1) = (1, 1)] = 2\Pr[X^0 = 0]\Pr[X^0 = 1]p_l(1 - q_l^2) + \\ + \{\Pr[X^0 = 1]\}^2(1 - q_l^3)^2$$

Pentru stările 2, …, k, variabilele X^{i-1}, Y^{i-1}, Z^{i-1}, W^{i-1} exprimă starea celor patru legături din etajul $i - 1$.

$$\Pr[(X^i, Y^i) = (u, v)] = \sum_{s_0 \ldots s_3 = 0,0,0,0}^{1,1,1,1} \Pr[(X^{i-1}, Y^{i-1}) = (s_0, s_1)] \cdot$$

$$\cdot \Pr[(Z^{i-1}, W^{i-1}) = (s_2, s_3)] \cdot \Pr[X^i = u \mid (X^{i-1}, Z^{i-1}) = (s_0, s_2)] \cdot$$

$$\cdot \Pr[Y^i = v \mid (Y^{i-1}, W^{i-1}) = (s_1, s_3)]$$

Probabilitătile conditionate sunt:

$$\Pr[X^i = 0 \mid (X^{i-1}, Z^{i-1}) = (0,0)] = 1$$

$$\Pr[X^i = 0 \mid (X^{i-1}, Z^{i-1}) = (0,1)] = q_l$$

$$\Pr[X^i = 0 \mid (X^{i-1}, Z^{i-1}) = (1,0)] = q_l$$

$$\Pr[X^i = 0 \mid (X^{i-1}, Z^{i-1}) = (1,1)] = q_l^2$$

Pentru etajul suplimentar $k + 1$:

$$\Pr[X^{k+1} = 0] = \Pr[(X^k, Y^k) = (0,0)] + \Pr[(X^k, Y^k) = (0,1)](q_l + q_l^2) +$$
$$+ \Pr[(X^k, Y^k) = (1,1)]q_l^3$$
$$\Pr[X^{k+1} = 1] = 1 - \Pr[X^{k+1} = 0]$$
$$\phi_r = p_r p_l \Pr[X^{k+1} = 1]$$

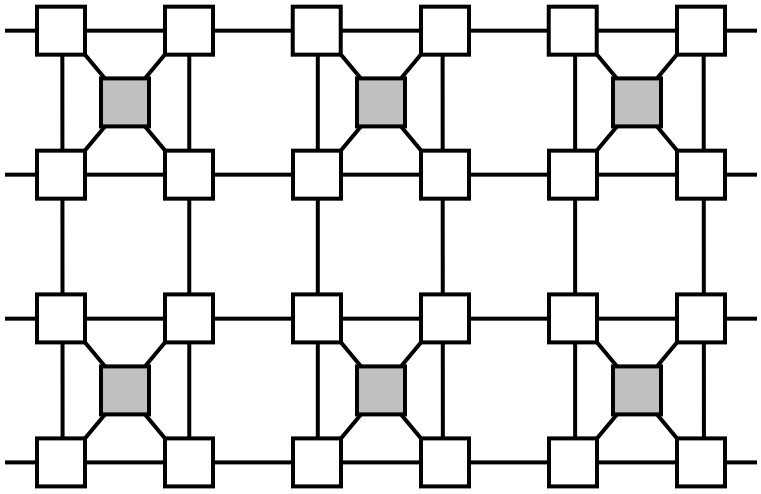
În final

$$A_r = N \phi_r$$

Măsura A_m se calculează similar înlocuind p_r cu p_m.

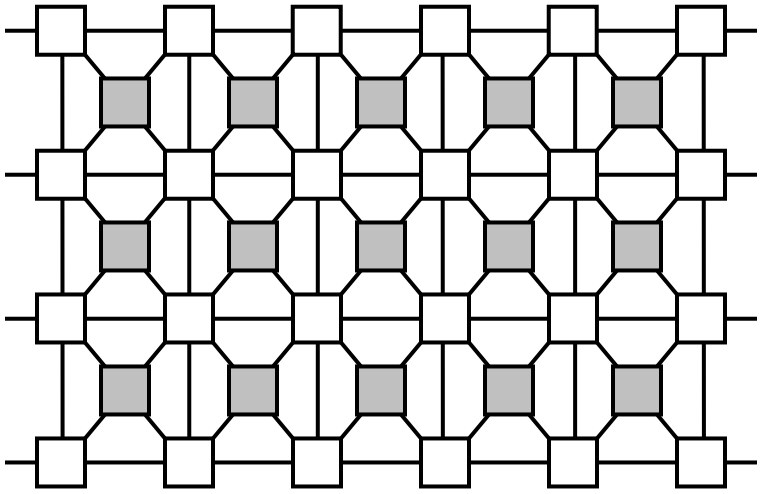
Plasa (mesh) interstitială

O retea conventională de genul plasă rectangulară bidimensională este incapabilă să tolereze vreun nod defect.
Redundanta interstitială (1, 4) este ilustrată de figura alăturată.



Nodurile umbrite sunt noduri de rezervă

Se observă câte un nod de rezervă adăugat pentru a înlocui oricare din cele patru noduri vecine care a clacat. Asadar, fiecare nod primar are un singur nod de rezervă, fiecare nod suplimentar este rezervă pentru patru noduri primare. Overheadul de redundantă este 25%.
Avantajul principal rezidă în proximitatea fizică a nodului de rezervă fată de nodurile primare pe care le poate înlocui. Aceasta poate reduce penalitatea de întârziere datorată utilizării unei rezerve.
Redundanta interstitială se practică uneori într-o formă diferită. Iată imediat, în figura alăturată, redundanta interstitială (4, 4).
Un nod primar are în această schemă patru noduri de rezervă si fiecare nod suplimentar este rezervă pentru patru noduri primare. O astfel de structură are un nivel de tolerantă mai ridicat dar si overheadul de redundantă este mai ridicat, este de cca. 100%.



Nodurile umbrite sunt noduri de rezervă

Fiabilitatea plasei cu redundantă interstitială (1, 4)

Plasa este de dimensiunile m×n, cu m si n numere pare. Reteaua este alcătuită din clustere de patru noduri primare cu un nod de rezervă. Reteaua (mesh) are în total $mn/4$ astfel de clustere.
Fie $R(t)$ fiabilitatea unui nod primar sau de rezervă.
Fiabilitatea unui cluster este

$$R_{cluster}(t) = R^5(t) + 5R^4(t)[1 - R(t)]$$

iar fiabilitatea unei plase (mesh) de m×n este

$$R_{plasa}(t) = [R_{cluster}(t)]^{mn/4}$$

Dacă pentru redundanta interstitială (1, 4) există această expresie pentru functia de fiabilitate, pentru schemele cu redundantă interstitială (4, 4) nu există un algoritm simplu pentru calculul fiabilitătii.

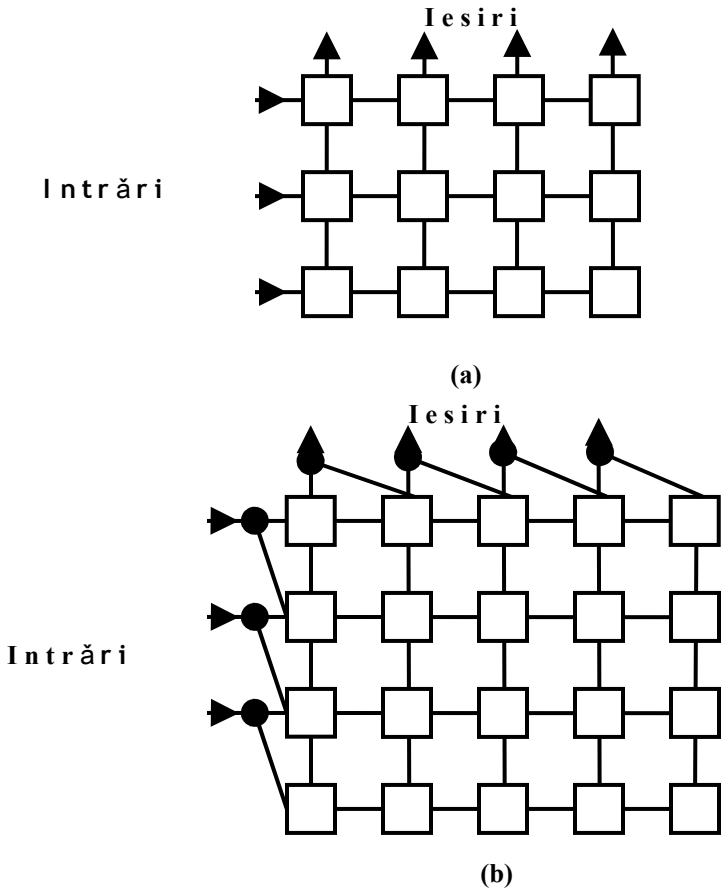
Retele crossbar fără redundante

Figura alăurată arată o retea crossbar 3x4 (trei intrări si patru iesiri). În general, o retea crossbar m×n are n intrări, m iesiri si mn comutatoare. Comutatoarele leagă toate perechile alcătuite dintr-o intrare si o iesire. Reteaua crossbar nu este tolerantă la defecte. Disfunctia oricărui comutator deconectează anumite perechi de noduri.

Retele crossbar cu redundante

Se adaugă redundante pentru a face reteaua tolerantă la defecte. Pentru aceasta se adaugă, de pildă o linie si o coloană de comutatoare. Conexiunile de intrare

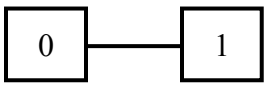
si de iesire sunt multiplicate prin faptul că fiecare intrare poate fi trimisă pe două linii si fiecare iesire poate fi obtinută de la două coloane.
Dacă un comutator se defectează, atunci linia si coloana de care apartine sunt înlocuite de linia si coloana de rezervă (v.figura).



Retele crossbar fără redundante (a) si cu redundante (b)

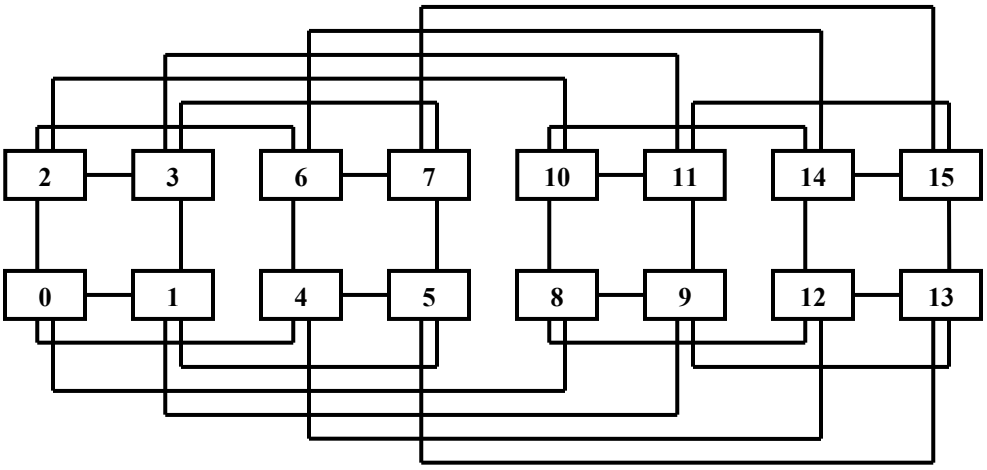
Retele de tip hipercub

Cu H_n se notează o retea de tip hipercub n-dimensională care ar 2^n noduri. Un hipercub 0-dimensional are un singur nod. Un hipercub H_n se construieste prin conectarea nodurilor corespondente din două retele H_{n-1}, hipercuburi cu o dimensiune mai putin. Muchiile adăugate pentru a conecta noduri corespunzătoare sunt numite muchii de dimensiune $(n - 1)$.



Muchie de dimensiune 0

Exemple de hipercuburi sunt date în figura alăturată. În hipercubul H_4 din figură se disting cu usurintă hipercuburi de dimensiuni inferioare si muchii de diverse dimensiuni.



Exemple diverse rezultate din lectura figurii:
Muchie de dimensiune 0: 8-9 (diferenta între numerele purtate de noduri: 1 = 2^0).
Muchie de dimensiune 1: 4-6 (diferenta între numerele purtate de noduri: 2 = 2^1).
Muchie de dimensiune 2: 10-14 (diferenta între numerele purtate de noduri: 4 = 2^2).
Muchie de dimensiune 4: 3-11 (diferenta între numerele purtate de noduri: 8 = 2^3).
Hipercuburi H_0: oricare nod.
Hipercuburi H_1: oricare pereche de noduri dintre următoarele: (0, 1), (2, 3), (4, 5), (6, 7), (8, 9), (10, 11), (12, 13), (14, 15).
Hipercuburi H_2: (0, 1, 2, 3), (4, 5, 6, 7), (8, 9, 10, 11) si (12, 13, 14, 15).
Hipercuburi H_3: (0, 1, 2, 3, 4, 5, 6, 7) si (8, 9, 10, 11, 12, 13, 14, 15).

Rutarea în hipercuburi

Pentru a simplifica rutarea se foloseste o numerotare specială. Numerele sunt exprimate în binar si dacă nodurile *i* si *j* sunt conectate de o muchie de dimensiune *k*, numerele pentru *i* si *j* diferă prin bitii de pe pozitia *k*.
Exemplu: nodurile 0000 si 0010 diferă numai prin bitul de pe pozitia 2^1; ele sunt conectate printr-o muchie de dimensiune 1.
Alt exemplu: un pachet trebuie să se deplaseze de la nodul $14 = 1110_2$ la nodul $2 = 0010_2$ într-o retea H_4. Rutările posibile sunt:

- 1110 → 0110 (dimensiune 3) → 0010 (dimensiune 2)

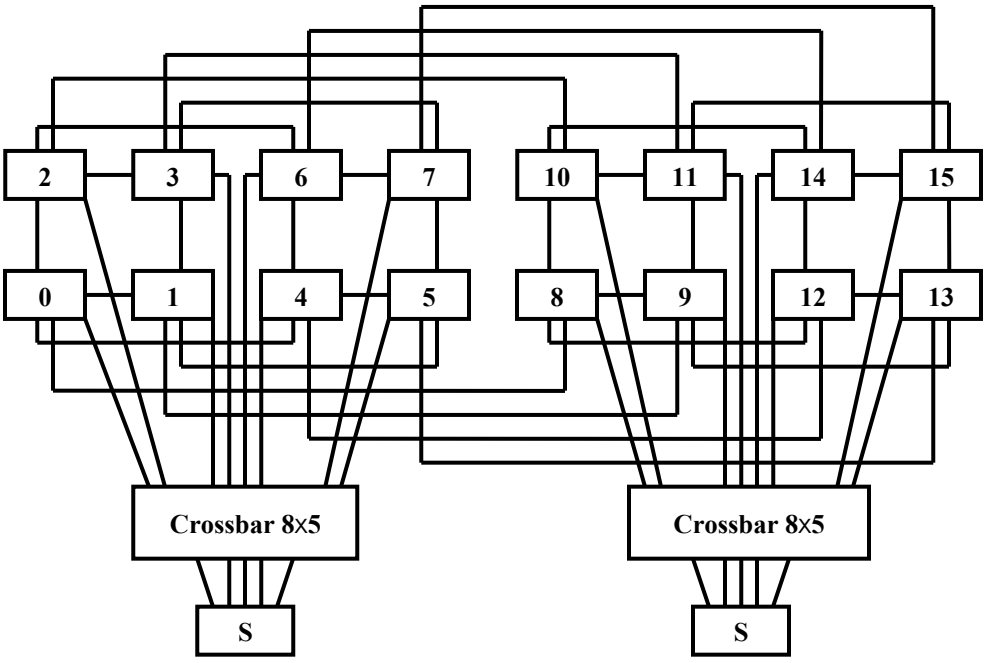
- 1110 → 1010 (dimensiune 2) → 0010 (dimensiune 3)

Rutarea în cazul general

Distanta între sursă si destinatie este în general numărul de biti diferiti în cele două adrese (distanta Hamming). Transferul de la nodul X la nodul Y poate fi făcut prin trecerea câte o dată pe fiecare din dimensiunile prin care sursa si destinatia diferă.
Dacă adresele sunt $X = x_{n-1} \ldots x_0$ si $Y = y_{n-1} \ldots y_0$, se definesc bitii $z_i = x_i \oplus y_i$ ($i = 0, \ldots, n-1$) cu $\oplus$ operatorul "sau-exclusiv".
Pachetul trebuie să traverseze o muchie în fiecare dimensiune pentru care $z_i = 1$.

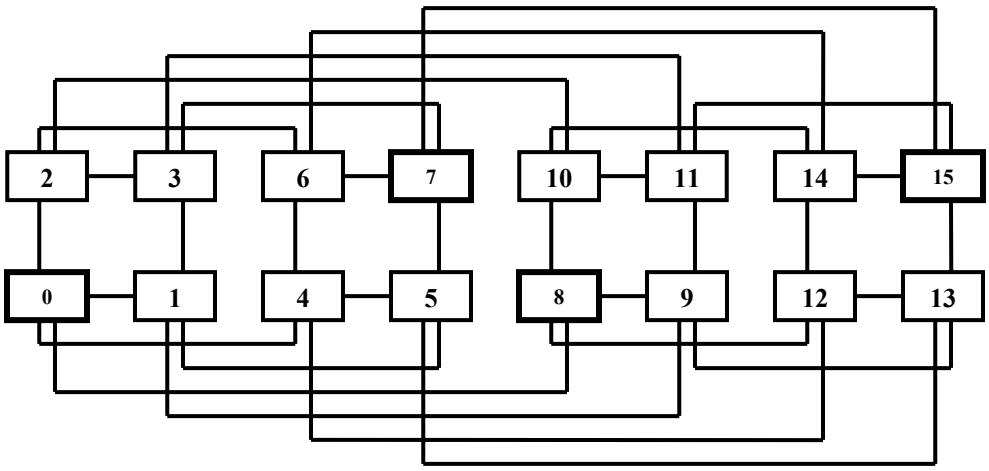
Toleranta la defecte în retelele hipercub

Pentru $n \geq 2$, un hipercub H_n poate tolera disfunctii ale legăturilor deoarece există căi multiple de la orice sursă la orice destinatie.
Disfunctiile nodurilor pot însă compromite operatia. O modalitate de ameliorare a situatiei constă în cresterea numărului de porturi de comunicare ale fiecărui nod de la n la $n + 1$ si conectarea acestor porturi suplimentare prin legături aditionale la unul sau mai multe noduri de rezervă.



Exemplu: Se pot adăuga două noduri de rezervă, fiecare din acestea fiind o rezervă pentru 2^{n-1} noduri ale unui subcub H_{n-1}.
Nodurile de rezervă ar putea necesita 2^{n-1} porturi. Numărul de porturi poate fi redus prin utilizarea unor comutatoare crossbar ale căror iesiri sunt conectate la

nodul de rezervă corespunzător. Numărul de porturi ale nodului de rezervă este redus la $n + 1$, acelasi ca pentru toate celelalte noduri.
Figura alăturată arată un hipercub H_4 cu două noduri de rezervă.
O metodă diferită de tolerare a defectelor constă în duplicarea procesoarelor din câteva (putine) noduri selectate. Fiecare procesor aditional este rezervă pentru oricare dintre procesoarele din nodurile vecine. În exemplul din figura următoare, nodurile 0, 7, 8, 15 ale unui hipercub H_4 sunt modificate prin duplicare (reprezentate îngrosat).



Fiecare nod are acum o rezervă la distantă nu mai mare de 1. Înlocuirea unui procesor defect cu unul din rezervă produce, desigur, o întârziere suplimentară în comunicare.

Rutarea în hipecuburi cu defecte

Algoritmul de rutare trebuie modificat pentru a ocoli nodurile sau legăturile defecte. Ideea de bază se poate formula astfel: se listează dimensiunile pe care un pachet trebuie să meargă si se parcurg acestea una câte una. Pe măsură ce muchiile sunt parcurse si marcate/eliminate (crossed off) din listă, dacă din cauza unui nod sau unei legături disfuncte legătura dorită nu este disponibilă se alege o altă muchie din listă (dacă este una) pentru continuarea parcursului; dacă pachetul atinge un anumit nod pentru a găsi toate dimensiunile din lista sa căzute, el revine (backtracks) la nodul anterior si încercarea continuă.
Algoritmul formal de rutare utilizează următoarele notatii:
TD – lista dimensiunilor pe care circulă mesajul, în ordinea parcurgerii.
TD^R – acelasi lucru în ordine inversă (reversed).
$\oplus_{i=1}^{k}$ – operatia sau-exclusiv executată de k ori, secvential.
Exemplu: $\oplus_{i=1}^{3} a_i$ înseamnă $(a_1 \oplus a_2) \oplus a_3$.

D – destinatie, S – sursă, $d = D \oplus S$ ($\oplus$ – operatia sau-exclusiv se execută bit-cu-bit pe bitii corespondenti din adresele binare D si S).
$SC(A)$ – multimea de noduri vizitată pe un parcurs pe fiecare din dimesiunile listate în multimea A.
Exemplu: la nodul 0010 – $SC(1, 3) = \{0000, 1000\}$.
e_n^i – un vector de n biti care are 1 în pozitia bitului i si 0 în rest.
Exemplu: $e_3^2 = 100$.
Pachetele sunt presupuse a consta în:

(i) d; $d = D \oplus S$
(ii) mesajul transmis ("încărcătura")
(iii) lista de dimensiuni vizitate deocamdată – TD

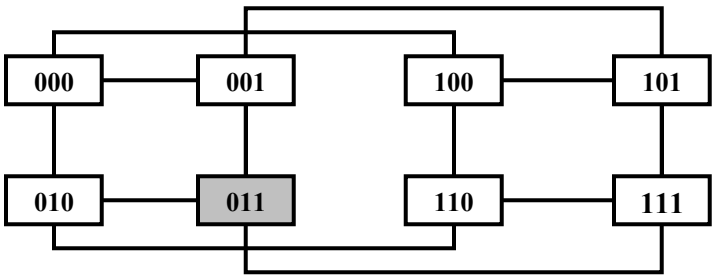
θ – operatia de adăgare (append). Scrierea $TD \; \theta \; x$ înseamnă adăugarea la finalul listei TD a lui x.
transmit(j) – rutina de trimitere a pachetului ($d \oplus e^j$, mesaj, $TD \; \theta \; x$) pe legătura j-dimensională de la nodul curent.

Algoritm de rutare pentru hipercuburi cu defecte

```
if (d == 0…0)
        destinatia a fost atinsă; exit
else
        for j = 0 to (n – 1) step 1 do {
                if (d_j == 1) && (legătura în dimensiunea j din acest nod este fără
                defect) && (e_v^j ∉ SC|TD^R) {
                transmit(j)
                exit
                }
        }
endif
if (există o legătură fără defect în SC|TD^R)
        fie h o astfel de legătură
else {
        g = max(m: ⊕_{i=1}^{m} e^{TD^R}(i) == 0...0 )
        if (g == numărul de elemente din SC(TD)) {
                nu există o cale
                exit
        else
                h = elementul al (g + 1)-lea din TD^R
        endif
        transmit(h)
end
```

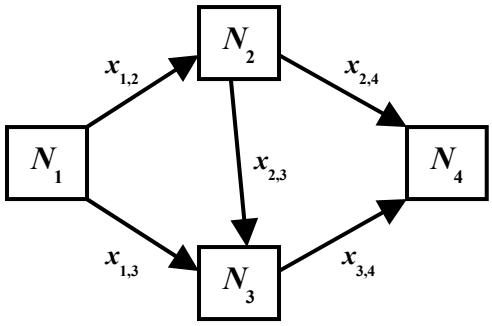
Exemplu pe hipercubul H_3:
H_3 cu nodul defect 011.

Nodul 000 trebuie să transmită un pachet la 111.
La 000, d = 111, trimite mesajul pe dimensiunea 0, la nodul 001.
La 001, d = 110 si TD = (0), tentative la muchiile de dimensiune 1: imposibil.
Bitul 2 din d este tot 1. Se verifică si si se stabileste că muchia de dimensiune 2 la 101 este disponibilă, mesajul este trimis la 101 si apoi la 111.
Exercitiu: Ce se întâmplă dacă sunt căzute nodurile 001 si 101?



Fiabilitatea retelelor punct-la-punct

Retelele nu sunt în mod necesar structuri regulate si de cele mai multe ori există mai multe căi între două noduri.
Se defineste *fiabilitatea terminală*: probabilitatea ca să existe o cale operațională între două noduri anumite, fiind date probabilitătile disfunctiilor pe fiecare legătură.
Exemplu: să se calculeze fiabilitatea terminală pentru perechea sursă-destinatie $N_1 - N_4$ (v.figura).



Sunt trei căi de la N_1 la N_4
- $P_1 = (x_{1,2}, x_{2,4})$
- $P_2 = (x_{1,3}, x_{3,4})$
- $P_3 = (x_{1,2}, x_{2,3}, x_{3,4})$

$p_{i,j}$ ($q_{i,j}$) – probabilitatea ca legătura $x_{i,j}$ să fie bună (respectiv defectă).

Nodurile sunt presupuse a fi fără defecte. Dacă nu aceasta este situatia, probabilitatea disfunctiei lor este încorporată în aceea a legăturilor care pleacă din noduri.
Multimea de căi trebue prelucrată pentru a obtine o multime echivalentă alcătuită din evenimente mutual exclusive, altminteri unele evenimente ar putea fi luate în calcul de mai multe ori.
Evenimente mutual exclusive în cazul în discutie:
(I) calea P_1 functională;
(II) calea P_2 functională si calea P_1 disfunctă;
(III) calea P_3 functională, căile P_1 si P_2 disfuncte.
Dacă numim reteaua din figură "punte" (denumire legată de forma si topologia ei), atunci fiabilitatea legăturii $N_1 - N_4$ este:

$$R_{\text{punte}} = p_{1,2}p_{2,4} + p_{1,3}p_{1,4}(1 - p_{1,2}p_{2,4}) + p_{1,2}p_{2,3}p_{3,4}(q_{1,3}q_{2,4})$$

Calculul fiabilităţii terminale

Pentru a calcula fiabilitatea terminală a unei retele cu m căi P_1, …, P_m de la sursă la destinatie se folosesc notatiile care urmează.
E_i ($\overline{E}_i$) – eveniment constând în operationalitatea (disfunctia) căii P_i.

R = Pr(existenta unei căi operationale) = $\Pr\left(\bigcup_{i=1}^{m} E_i\right)$.

Multimea de evenimente poate fi descompusă în evenimente mutual exclusive. După descompunere, expresia evenimentului "există o cale operațională" este

$$E_1 \cup (E_2 \cap \overline{E}_1) \cup (E_3 \cap \overline{E}_1 \cap \overline{E}_2) \cup \ldots \cup (E_m \cap \overline{E}_1 \cap \ldots \overline{E}_{m-1})$$

si

$$R = \Pr(E_1) + \Pr(E_2 \cap \overline{E}_1) + \Pr(E_3 \cap \overline{E}_1 \cap \overline{E}_2) + \ldots$$
$$\ldots + \Pr(E_m \cap \overline{E}_1 \cap \ldots \cap \overline{E}_{m-1})$$

Expresia din urmă poate fi rescrisă uzând de probabilităti conditionate

$$R = \Pr(E_1) + \Pr(E_2)\Pr(\overline{E}_1|E_2) + \Pr(E_3)\Pr(\overline{E}_1 \cap \overline{E}_2|E_3) + \ldots$$
$$\ldots + \Pr(E_m)\Pr(\overline{E}_1 \cap \ldots \cap \overline{E}_{m-1}|E_m)$$

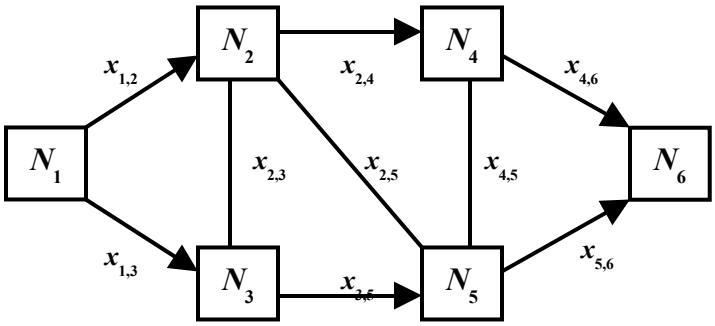
Problema centrală este calcularea probabilităților conditionate de forma generală $\Pr(\overline{E}_1 \cap \ldots \cap \overline{E}_{i-1}|E_i)$.
Pentru a identifica legăturile care trebuie să cadă pentru ca E_i să aibă loc dar nu E_1, …, E_{i-1}, se folosesc asa-numitele multimi conditionate

$$S_{j/i} = P_j - P_i = \{x | x \in P_j \text{ si } x \notin P_i\}$$

Identificarea evenimentelor disjuncte în cazul general nu este totdeauna o treabă facilă.

Exemplu suplimentar pentru fiabilitatea terminală

O retea cu sase noduri care are 9 legături unidirectionale si 3 bidirectionale.



O listă cu toate căile de la N_1 la N_6:

$P_1 = \{x_{1,3}, x_{3,5}, x_{5,6}\}$
$P_2 = \{x_{1,2}, x_{2,5}, x_{5,6}\}$
$P_3 = \{x_{1,2}, x_{2,4}, x_{4,6}\}$
$P_4 = \{x_{1,3}, x_{3,5}, x_{4,5}, x_{4,6}\}$
$P_5 = \{x_{1,3}, x_{2,3}, x_{2,4}, x_{4,6}\}$
$P_6 = \{x_{1,3}, x_{2,3}, x_{2,5}, x_{5,6}\}$
$P_7 = \{x_{1,2}, x_{2,5}, x_{4,5}, x_{4,6}\}$
$P_8 = \{x_{1,2}, x_{2,3}, x_{3,5}, x_{5,6}\}$
$P_9 = \{x_{1,2}, x_{2,4}, x_{4,5}, x_{5,6}\}$
$P_{10} = \{x_{1,3}, x_{2,3}, x_{2,4}, x_{4,5}, x_{5,6}\}$
$P_{11} = \{x_{1,3}, x_{2,3}, x_{2,5}, x_{4,5}, x_{4,6}\}$
$P_{12} = \{x_{1,3}, x_{3,5}, x_{2,5}, x_{2,4}, x_{4,6}\}$
$P_{13} = \{x_{1,3}, x_{2,3}, x_{3,5}, x_{4,5}, x_{4,6}\}$

Căile, se observă, sunt ordonate de la cea mai scurtă la cea mai lungă.

Pentru a calcula alti termeni din sumă, trebuie avută în vedere intersectia mai multor multimi conditionate.

$P_1 = \{x_{1,3}, x_{3,5}, x_{5,6}\}$
$P_2 = \{x_{1,2}, x_{2,5}, x_{5,6}\}$
$P_3 = \{x_{1,2}, x_{2,4}, x_{4,6}\}$
$P_4 = \{x_{1,3}, x_{3,5}, x_{4,5}, x_{4,6}\}$

Pentru a calcula termenul al patrulea – expresia lui P_4 – multimile conditionate sunt:

$$S_{1/4} = \{x_{5,6}\};\ S_{2/4} = \{x_{1,2}, x_{2,5}, x_{5,6}\};\ S_{3/4} = \{x_{1,2}, x_{2,4}\}$$

$S_{1/4}$ este inclus în $S_{2/3}$; dacă $S_{1/4}$ este cu defecte, atunci si $S_{2/4}$ este cu defecte. $S_{2/4}$ poate fi ignorat în acest caz.

Al patrulea termen din ecuatia fiabilitătii este

$$p_{1,3}p_{3,5}p_{4,5}p_{4,6}(1 - p_{5,6})(1 - p_{1,2}p_{2,4})$$

Calculul termenului al treilea conduce la

$$S_{1/3} = \{x_{1,3}, x_{3,5}, x_{5,6}\}$$
$$S_{2/3} = \{x_{2,5}, x_{5,6}\}$$

Cele două multimi conditionate nu sunt disjuncte.

Evenimentul care constă în defectarea simultană a multimilor de arce $S_{1/3}$ si $S_{2/3}$ trebuie să fie împărtit în evenimente disjuncte:

(I) $x_{5,6}$ cu defecte

(II) $x_{5,6}$ este operational si atât $x_{1,3}$ cât si $x_{2,5}$ sunt defecte

(III) atât $x_{1,3}$ cât si $x_{5,6}$ sunt în functiune si atât $x_{2,5}$ cât si $x_{3,5}$ sunt defecte.

Pentru termenul al treilea rezultă expresia

$$p_{1,2}p_{2,4}p_{4,6}(q_{5,6} + p_{5,6}q_{1,3}q_{2,5} + p_{5,6}p_{1,3}q_{2,5}q_{3,5})$$

Termenii rămasi se calculează similar.
Fiabilitatea terminală este suma tuturor celor 13 termeni definiti mai devreme.

SISTEME DE DISCURI TOLERANTE LA DEFECTE

Pentru a spori siguranta în functionare, sistemele de memorii pot fi structurate în asa manieră încât prin utilizarea unor redundante ele să manifeste o anumită tolerantă la defecte. În alti termeni, prin măsuri prealabile anumite subsisteme pot suplini alte subsisteme, care dintr-un motiv sau altul se defectează. Iată în continuare un exemplu semnificativ în ceea ce priveste toleranta la defecte a sistemelor de memorare, în particular memorarea pe disc.

Memorii ieftine exploatate în conditii de sigurantă

Fie n dispozitive de memorare, D_1, D_2, …, D_n. Fiecare dintre ele contine k octeti si sunt dispozitive de stocat date. Fie alte m dispozitive de memorare C_1, C_2, …, C_m. Si acestea contin fiecare tot câte k octeti si sunt denumite dispozitive de verificare. Continutul fiecărui dispozitiv de verificare se calculează din continutul dispozitivelor de date. Problema este a calcula continutul dispozitivelor C_i în asa mod încât oricare m dispozitive din D_1, D_2, …, D_n, C_1, C_2, …, C_m s-ar defecta, continutul dispozitivelor defecte să poată fi reconstituit din continutul dispozitivelor în functie.

Strategia generală

Formal, modelul defectului este acela al unei pierderi de informatie prin stergere. Dacă un dispozitiv se defectează el iese din joc si sistemul recunoaste această situatie de inutilitate. Pierderea aceasta diferă de aparitia unor erori, caz în care defectarea se manifestă prin stocarea si restituirea unor valori incorecte care pot fi recunoscute printr-un anume gen de codare intrinsecă.
Calculul continutului fiecărui dispozitiv de verificare C_i necesită o functie F_i aplicată tuturor dispozitivelor de date. Formulele următoare sunt un exemplu pentru $n = 8$ si $m = 2$. Continutul dispozitivelor de verificare C_1 si C_2 se obtine prin evaluarea functiilor F_1, respectiv F_2.

$$C_1 = F_1(D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8)$$
$$C_2 = F_2(D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8)$$

Metoda de codare RS-RAID (RS – de la Reed-Solomon, RAID – de la Redundant Arrays of Inexpensive/Independent Disks) divide fiecare dispozitiv de memorare în cuvinte. Fiecare cuvânt este alcătuit din w biti, număr ales de programator dar raportat la anumite restrictii. Asadar, fiecare dispozitiv contine

$$l = (k \text{ octeti})\left(\frac{8 \text{ biti}}{\text{octet}}\right)\left(\frac{1 \text{ cuvânt}}{w \text{ biti}}\right) = \frac{8k}{w} \text{ cuvinte}$$

Functiile de codare F_i operează pe cuvinte cu rezultatul tot în cuvinte, ca în relatiile următoare, unde $x_{i,j}$ reprezintă cuvântul j din dispozitivul de memorare X_i.

D_1	D_2	C_1	C_2
$d_{1,1}$	$d_{2,1}$	$c_{1,1} = F_1(d_{1,1}, d_{2,1})$	$c_{2,1} = F_2(d_{1,1}, d_{2,1})$
$d_{1,2}$	$d_{2,2}$	$c_{1,2} = F_1(d_{1,2}, d_{2,2})$	$c_{2,2} = F_2(d_{1,2}, d_{2,2})$
$d_{1,3}$	$d_{2,3}$	$c_{1,3} = F_1(d_{1,3}, d_{2,3})$	$c_{2,3} = F_2(d_{1,3}, d_{2,3})$
.	.	.	.
$d_{1,i}$	$d_{2,i}$	$c_{1,i} = F_1(d_{1,i}, d_{2,i})$	$c_{2,i} = F_2(d_{1,i}, d_{2,i})$

Pentru notatii mai simple, cu un indice mai putin, se admite că fiecare dispozitiv retine un cuvânt si numai unul. Pe calea aceasta problema se reduce la n cuvinte-date, $d_1, d_2, \ldots, d_n$ si la m cuvinte de verificare $c_1, c_2, \ldots, c_m$ calculate din cuvintele-date în asa mod încât pierderea oricăror m cuvinte să fie tolerată. Pentru calculul unui cuvânt de verificare c_i depus în dispozitivul C_i se aplică functia F_i cuvintelor-date

$$c_i = F_i(d_1, d_2, ..., d_n)$$

Dacă un cuvânt-dată din dispozitivul D_j este actualizat de la d_j la d_j' atunci fiecare din cuvintele de verificare c_i trebuie recalculat prin utilizarea unei functii $G_{i,j}$ astfel încât

$$c_i' = G_{i,j}(d_j, d_j', c_i)$$

Când m dispozitive de memorare clachează se reconstruieste sistemul după cum urmează. Mai întâi, pentru fiecare dispozitiv defect D_j se construieste o functie care să recupereze continutul lui D_j din cuvintele depuse în dispozitivele functionale. Când operatia aceasta este încheiată se reconstituie continutul unor eventuale dispozitive de verificare disfuncte C_i, cu ajutorul functiilor F_i.

De exemplu, presupunând că $m = 1$, paritatea $n + 1$ se poate descrie în termenii generali de mai sus. Există numai un dispozitiv de verificare C_1 si lungimea cuvântului este de 1 bit ($w = 1$). Pentru calculul cuvâtului de verificare c_1 se ia paritatea prin SAU EXCLUSIV (XOR) a cuvintelor de date

$$c_1 = F_1(d_1, d_2, ..., d_n) = d_1 \oplus d_2 \oplus ... \oplus d_n$$

Dacă cuvântul de pe suportul D_j se schimbă din d_j în d_j' atunci c_1 se recalculează din paritatea vechiului cuvânt si din cele două cuvinte-date

$$c_1' = G_{1,j}(d_j, d_j', c_1) = c_1 \oplus d_j \oplus d_j'$$

Dacă un dispozitiv se defectează atunci fiecare cuvânt poate fi reconstituit prin paritate a cuvintelor de pe dispozitivele rămase în functie

$$d_j = d_1 \oplus ... \oplus d_{j-1} \oplus d_{j+1} \oplus ... \oplus d_n \oplus c_1$$

Sistemul este rezistent la defectarea oricărui (unic) suport.

O reformulare a problemei sună astfel: sunt date n date d_1, d_2, …, d_n, toate de dimensiunea w. Se definesc functiile F si G care sunt utilizate pentru a calcula si pentru a întretine, a mentine actuale cuvintele de verificare c_1, c_2, …, c_m. Se face o descriere a modului cum se reconstituie cuvintele pe orice suport esuat când numărul de dispozitive de memorare defecte nu depăseste m. Odată cuvintele-date reconstituite se recalculează cuvintele de verificare din cuvintele-date cu ajutorul functiilor F. Sistemul este refăcut în întregime.

Algoritmul RS-RAID

Trei aspecte sunt deosebite în aplicarea algoritmului. Primul constă în utilizarea matricilor Vandermonde (Alexandre-Théophile Vandermonde, 1735-1796) pentru calculul si mentinerea cuvintelor de control, al doilea este utilizarea eliminării Gauss pentru recuperarea din starea de nefunctionare si al treilea, utilizarea aritmeticii specifice câmpurilor Galois. Acestea toate sunt detaliate în continuare.

Calculul si întretinerea cuvintelor de verificare

Functiile F_i sunt prin definitie combinatii liniare ale cuvintelor-date

$$c_i = F_i(d_1, d_2, ..., d_n) = \sum_{j=1}^{n} d_j f_{i,j}$$

Cu alte cuvinte, dacă se adoptă o reprezentare matricială cu D si C vectori si F_i linii într-o matrice F

$$FD = C$$

Matricea F este definită ca o matrice Vandermonde $m \times n$ cu $f_{i,j} = j^{i-1}$, ceea ce face din relatia de mai sus

$$\begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,n} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,n} \\ \vdots & \vdots & & \vdots \\ f_{m,1} & f_{m,2} & \cdots & f_{m,n} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 3 & \cdots & n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 2^{m-1} & 3^{m-1} & \cdots & n^{m-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}$$

Când unul din cuvintele-date d_j se schimbă în d_j' cuvintele de verificare trebuie schimbate în consecintă. Prin scăderea portiunii din cuvântul de verificare care corespunde lui d_j si adunarea cantitătii necesare pentru d_j' se obtine pentru $G_{i,j}$ definitia din relatia de mai jos

$$c_i' = G_{i,j}(d_j, d_j', c_i) = c_i + f_{i,j}(d_j' - d_j)$$

Asadar, calcularea si întretinerea cuvintelor de verificare pot fi făcute printr-o aritmetică simplă, dar după regulile date mai departe.

Recuperarea din *crash*

Pentru a explica recuperarea aceasta, se definesc matricea $A = \begin{bmatrix} I \\ F \end{bmatrix}$ si vectorul $E = \begin{bmatrix} D \\ C \end{bmatrix}$. Apoi se scrie ecuatia $AD = E$ care în formă detaliată apare ca

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 3 & \cdots & n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 2^{m-1} & 3^{m-1} & \cdots & n^{m-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \\ c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}$$

Se observă că fiecare suport din sistem are o linie în matricea A si în vectorul E. Dacă un dispozitiv esuează esecul se materializează în relatia de mai sus prin ignorarea/stergerea liniei care corespunde acelui dispozitiv. Rezultă o matrice A' si un vector E' cu linii mai putine dar care verifică o ecuatie asemănătoare cu cea de mai sus

$$A'D = E'$$

Dacă exact m dispozitive sunt inutilizabile atunci matricea A' este o matrice $n \times n$. Deoarece matricea F este de tipul Vandermonde orice submultime de linii ale ei este liniar independentă. Matricea A' este, asadar, nesingulară si valorile care compun vectorul D pot fi calculate din ecuatia matricială de mai sus prin eliminare Gauss. Prin urmare toate datele pot fi recuperate.

Cu D odată obtinut, valorile oricărui suport cu date de verificare C_i esuat pot fi si ele reconstituite. Dacă sunt mai putin de m dispozitive cu probleme, alegerea la întâmplare a unui număr de exact n linii din A' permite eliminarea gaussiană si continuarea este evidentă. Sistemul tolerează până la m dispozitive inutilizabile.

Aritmetica în câmpurile Galois

O preocupare majoră în algoritmul RS-RAID o constituie domeniul calculelor care este o multime de cuvinte binare de lungime fixă w. Recuperarea dintr-o eroare comisă obisnuit ar putea consta în efectuarea unor calcule modulo 2^w. Maniera aceasta însă nu functionează deoarece împărtirea nu-i definită pentru orice pereche de elemente. De exemplu, 3:2 *modulo* 4 nu este definită. Această situatie face eliminarea Gauss imposibilă în foarte multe cazuri.

Câmpurile cu 2^w elemente sunt câmpuri Galois – notate $GF(2^w)$ – un subiect fundamental în algebra abstractă. Mai jos se definesc moduri eficiente de a aduna, scădea, multiplica si împărti elemente apartinând unui câmp Galois.

Elementele unui câmp Galois $GF(2^w)$ sunt întregi de la zero la $2^w - 1$. Adunarea si scăderea sunt aplicatii simple: operatii XOR (SAU EXCLUSIV). De pildă, în $GF(2^4)$

$$11 + 7 = 1011 \oplus 0111 = 1100 = 12$$
$$11 - 7 = 1011 \oplus 0111 = 1100 = 12$$

Multiplicarea si diviziunea sunt mai complexe. Când w este mic, 16 sau mai mic, se utilizează tabele de logaritmi lungi de $2^w - 1$. Tabelul contine indici, o functie *gflog* si o functie *gfilog*. Ambele functii sunt functii cu valori întregi. Prima este listată pentru indici de la 1 la $2^w - 1$ si este o listă de logaritmi în câmpul Galois, a doua este definită pentru indici de la 0 la $2^w - 2$ si contine rezultatul operatiei inverse logaritmării. Evident, compunerea celor două functii, în orice ordine produce functia identitate, $gflog[gfilog(i)] = i$, $gfilog[gflog(i)] = i$. Cu aceste functii se pot executa operatiile de multiplicare si de împărtire prin luarea logaritmilor factorilor, prin calculul sumei sau a diferentei valorilor obtinute (ca elemente ale câmpului) si revenirea la rezultat prin operatia inversă. Iată un tabel de logaritmi pentru $w = 4$.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
gflog(i)		0	1	4	2	8	5	10	3	14	9	7	6	13	11	12
gfilog(i)	1	2	4	8	3	6	12	11	5	10	7	14	15	13	9	

Evident, numai numerele nenule au logaritmi. Logaritnul invers al unui număr i este egal cu logaritmul invers al numărului $[i \bmod (2^w - 1)]$.
Exemple de calcule în aritmetica din $GF(2^4)$:

$$3*7 = gfilog[gflog(3) + gflog(7)] = gfilog[4 + 10] = gfilog[14] = 9$$
$$13*10 = gfilog[gflog(13) + gflog(10)] = gfilog[13 + 9] = gfilog[7] = 11$$
$$13/10 = gfilog[gflog(13) - gflog(10)] = gfilog[13 - 9] = gfilog[4] = 3$$
$$3/7 = gfilog[gflog(3) - gflog(7)] = gfilog[4 - 10] = gfilog[9] = 10$$

Asadar, o multiplicare sau o diviziune necesită trei apeluri la tabel – două pentru logaritmi si unul pentru inversul logaritmului –, o adunare sau o scădere si o operatie de tip *modulo*.
Aritmetica unui câmp Galois are fundamentele date în continuare.
Un câmp $GF(n)$ este o multime de n elemente închisă la operatiile de adunare si multiplicare, cu un invers (opus) raportat la adunare pentru fiecare element, cu un invers raportat la operatia de înmultire pentru fiecare element nenul. Câmpul $GF(2)$ de exemplu, contine două elemente, adunarea si înmultirea se practică *modulo* 2 (operatorii XOR si AND, respectiv). Analog, dacă n este prim atunci $GF(n)$ este multimea $\{0, 1, \ldots, n - 1\}$ în care adunarea si înmultirea se practică *modulo n*.
Dacă $n > 1$ nu este prim, atunci multimea $\{0, 1, \ldots, n - 1\}$ cu adunarea si multiplicarea *modulo n* nu este câmp. De exemplu, dacă $n = 4$ atunci multimea $\{0, 1, 2, 3\}$, închisă la adunare si înmultire nu este câmp deoarece 2 nu are un invers la înmultire, nu există a astfel încât $2a = 1 \pmod 4$. Asadar, nu se poate

face codarea cu cuvinte binare de lungime $w > 1$ cu operatiile de adunare si înmultire modulo 2^w. În loc trebuie utilizat câmpul Galois corespunzător.
Explicatiile relative la câmpurile Galois uzează de polinoamele într-o nedeterminată cu coeficienti în $GF(2)$. Adică, dacă $r(x) = x + 1$ si $s(x) = x$ atunci $r(x) + s(x) = 1$ doarece $x + x = (1 + 1)x = 0x = 0$. Mai mult, se pot lua polinoame de acest gen *modulo* alte polinoame conform cu identitatea: dacă $r(x)$ mod $q(x) = s(x)$, atunci $s(x)$ este un polinom de grad inferior lui $q(x)$ si $r(x) = q(x)t(x) + s(x)$ cu $t(x)$ un polinom în x.
Dacă, de exemplu, $r(x) = x^2 + x$ si $q(x) = x^2 + 1$ atunci $r(x)$ mod $q(x) = x + 1$.
Fie acum $q(x)$ un polinom *primitiv* de gradul w cu coeficienti în $GF(2)$. Primitiv înseamnă că polinomul nu are divizori cu coeficienti în $GF(2)$ si polinomul x este generatorul câmpului $GF(2^w)$. Cum generează x câmpul? Se consideră initial elementele obligatorii 0, 1 si x si se continuă enumerarea elementelor obtinute prin multiplicarea ultimului element cu x si retinerea rezultatului modulo $q(x)$. Enumerarea se încheie cu elementul pentru care rezultatul *modulo* $q(x)$ este egal cu 1.
Dacă, de exemplu, $w = 2$ si $q(x) = x^2 + x + 1$ atunci primele elemente sunt 0, 1 si x, iar x^2 mod $q(x) = x + 1$ si cele patru elemente ale câmpului $GF(4)$ sunt $\{0, 1, x, x + 1\}$. Un al cincilea element nu există deoarece $x(x + 1) = x^2 + x$ care luat *modulo* $q(x)$ produce 1, element deja existent.
Câmpul general $GF(2^w)$ se construieste prin găsirea unui polinom primitiv $q(x)$ de gradul w peste $GF(2)$ urmată de enumerarea elementelor generate de x. Adunarea si multiplicarea elementelor câmpului se fac după regulile adunării si multiplicării polinoamelor cu grija de a lua totdeauna rezultatul *modulo* $q(x)$. Un asemenea câmp se mai scrie ca $GF(2^w) = GF(2)[x]/q(x)$.
Acum, pentru a uza de un câmp $GF(2^w)$ în algoritmul RS-RAID este necesară definirea unei aplicatii a elementelor din acest câmp pe cuvinte binare de lungime w. Un polinom $r(x)$ din $GF(2^w)$ poate fi aplicat pe un cuvânt binar b de lungime w prin punerea celui de al i-lea bit din b egal cu coeficientul puterii x^i din polinom. Pentru $GF(4) = GF(2)[x]/x^2 + x + 1$ se obtine tabelul următor

Elemente generate	Elemente polinomiale	Elemente binare	Reprezentarea zecimală
0	0	00	0
x^0	1	01	1
x^1	x	10	2
x^2	$x + 1$	11	3

Adunarea elementelor se realizează prin operatia SAU EXCLUSIV (XOR) bit cu bit. Multiplicarea este mai complicată: se iau elementele sub forma polinomială, se multiplică ca polinoame si se ia rezultatul *modulo* $q(x)$. Tabelele de logaritmi ca acela de mai sus se bazează pe o tabelă de compunere ca aceea dată pentru cazul $GF(4)$.

Pentru alte valori w, polinoame primitive $q(x)$ se găsesc în literatură. Iată câteva:

$w = 4$: $x^4 + x + 1$
$w = 8$: $x^8 + x^4 + x^3 + x^2 + 1$
$w = 16$: $x^{16} + x^{12} + x^3 + x + 1$
$w = 32$: $x^{32} + x^{22} + x^3 + x + 1$
$w = 64$: $x^{64} + x^4 + x^3 + x + 1$

Cu elementul de pornire $x^0 = 1$, $GF(2^w)$ se completează prin enumerarea elementelor obtinute prin multiplicarea cu x a ultimului element enumerat si luarea rezultatelor *modulo* $q(x)$. Tabelul care urmează cuprinde cazul câmpului $GF(2^4)$ cu polinomul primitiv $q(x) = x^4 + x + 1$. În acelasi tabel se observă si modul cum se generează tabelele de logaritmi si de invers-logaritmi prezentate mai devreme.

Element generat	Polinom	Exprimare binară	Exprimare zecimală
0	0	0000	0
x^0	1	0001	1
x^1	x^1	0010	2
x^2	x^2	0100	4
x^3	x^3	1000	8
x^4	$x + 1$	0011	3
x^5	$x^2 + x$	0110	6
x^6	$x^3 + x^2$	1100	12
x^7	$x^3 + x + 1$	1011	11
x^8	$x^2 + 1$	0101	5
x^9	$x^3 + x$	1010	10
x^{10}	$x^2 + x + 1$	0111	7
x^{11}	$x^3 + x^2 + x$	1110	14
x^{12}	$x^3 + x^2 + x + 1$	1111	15
x^{13}	$x^3 + x^2 + 1$	1101	13
x^{14}	$x^3 + 1$	1001	9
x^{15}	1	0001	1

Sumarul algoritmului

Fiind date n dispozitive pentru date si m dispozitive de control, algoritmul RS-RAID care le face tolerante la cel mult m defecte se aplică următoarea secventă de operatii:

1. Se alege o valoare pentru w astfel ca $2^w > m + n$. Este convenabil a se alege $w = 8$ sau $w = 16$, ceea ce conduce la cuvinte numărate în octeti (bytes). Pentru $w = 16$ suma $m + n$ poate fi până la 65535.

2. Se stabilesc tabelele cu functiile *gflog* si *gfilog* după metoda dată mai devreme
3. Se calculează matricea F care este o matrice Vandermonde $m \times n$: $f_{i,j} = j^{i-1}$ (pentru $1 \le i \le m$, $1 \le j \le n$) cu operatii în $GF(2^w)$.
4. Matricea F se foloseste la calculul si la întretinerea dispozitivelor de verificare, din cuvinte depuse pe dispozitivele de date. Iarăsi, operatiile se fac în $GF(2^w)$.
5. Dacă un număr de dispozitive, mai putine de m clachează atunci ele se reconstituie în maniera care urmează. Se aleg oricare n dispozitive din cele rămase în functie si se construiesc matricea *A'* si vectorul *E'* ca mai sus. Se rezolvă apoi pentru D ecuatia $A'D = E'$. Prin aceasta datele de pe dispozitivele de stocare a datelor sunt recuperate. Acum se pot reconstitui si dispozitivele de verificare esuate, prin utilizarea matricii F.

Un exemplu. Se presupune că sunt trei suporturi de date si trei suporturi de verificare si fiecare dintre ele detine un megaoctet. Asadar, $n = 3$ si $m = 3$. Se alege $w = 4$, asa încât $2^w > m + n$. Pentru multiplicări se foloseste tabelul dat mai devreme pentru $GF(2^4)$. În aceste conditii matricea F este

$$F = \begin{bmatrix} 1^0 & 2^0 & 3^0 \\ 1^1 & 2^1 & 3^1 \\ 1^2 & 2^2 & 3^2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 4 & 5 \end{bmatrix}$$

Se pot calcula acum cuvintele de verificare prin relatia $C = FD$. Se admite că primele cuvinte stocate pe cele trei dispozitive de date sunt, respectiv, 3, 13, 9. Calculul cuvintelor de control C_1, C_2, C_3 produce valorile următoare:

$$C_1 = 1*3 + 1*13 + 1*9 = 3 + 13 + 9 = 0011 \oplus 1101 \oplus 1001 = 0111 = 7$$

$$C_2 = 1*3 + 2*13 + 3*9 = 3 + 9 + 8 = 0011 \oplus 1001 \oplus 1000 = 0010 = 2$$

$$C_3 = 1*3 + 4*13 + 5*9 = 3 + 1 + 11 = 0011 \oplus 0001 \oplus 1011 = 1001 = 9$$

Dacă, de pildă, continutul dispozitivului de date cu indicele 2 se modifică si primul număr devine 1, atunci fiecare din dispozitivele de verificare primeste valoarea $(1 - 13) = (0001 \oplus 1101) = 1100 = 12$, care este utilizată pentru recalcularea valorilor de verificare

$$C_1 = 7 + 1*12 = 0111 \oplus 1100 = 1011 = 11$$

$$C_2 = 2 + 2*12 = 2 + 11 = 0010 \oplus 1011 = 1001 = 9$$

$$C_3 = 9 + 4*12 = 9 + 5 = 1001 \oplus 0101 = 1100 = 12$$

Dacă D_2, D_3 si C_3 se pierd atunci, din matricea A si din vectorul E se sterg liniile care corespund dispozitivelor defecte pentru a obtine ecuatia $A'D = E'$

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix} D = \begin{bmatrix} 3 \\ 11 \\ 9 \end{bmatrix}$$

Prin eliminare Gauss se poate inversa matricea *A'* si se obtine

$$D = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 3 & 1 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 11 \\ 9 \end{bmatrix}$$

si valorile reconstituite sunt

$$D_2 = 2*3 + 3*11 + 1*9 = 6 + 14 + 9 = 1$$

$$D_3 = 3*3 + 2*11 + 1*9 = 5 + 5 + 9 = 9$$

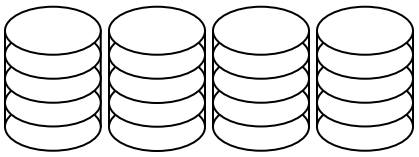
Cu matricea *F* se poate reconstitui si

$$C_3 = 1*3 + 4*1 + 5*9 = 3 + 4 + 11 = 12$$

si sistemul este recuperat în întregime.
Există si alte sisteme RAID la care se fac scurte referiri în continuare.

RAID nivelul 0 (fără redundantă)

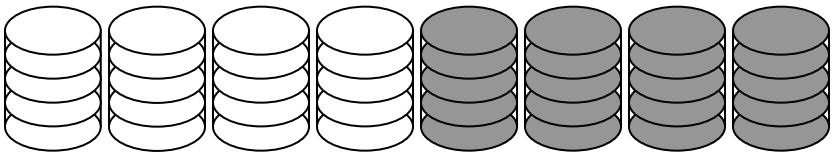
Un sistem de discuri non-redundant (sau de nivel 0) are costul cel mai scăzut din cauza lipsei oricărei redundante. Schema aceasta oferă cea mai bună performantă la scriere deoarece nu necesită vreodată actualizarea vreunei informatii redundante. Surpinzător, nu are cea mai bună performantă la citire. Schemele redundante (ca aceea numită "în oglindă" sau "oglindită", care creează duplicate ale datelor) pot executa mai bine citirile prin planificarea selectivă a cererilor pe discul cu timpul de căutare mediu cel mai scurt si întârzierea rotativă cea mai mică. Fără redundante, orice cădere a unui disc va produce pierderi de date. Sistemele de discuri fără redundante sunt larg utilizate în supercalcul unde performanta si capacitatea trec ca importantă înaintea fiabilitătii.



Sistem de discuri fără redundante

RAID nivelul 1 (sisteme "în oglindă")

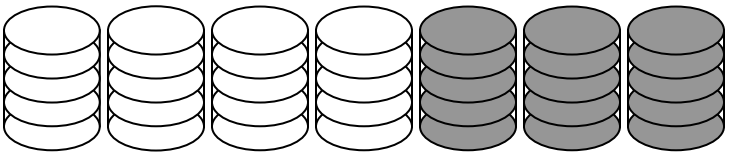
Sistemul traditional denumit "în oglindă" sau "cu umbră" utilizează de două ori mai multe discuri decât un sistem de discuri fără redundante. Ori de câte ori un articol-dată este scris pe un disc el este scris si pe un disc redundant, astfel există totdeauna două cópii ale informatiei, două exemplare. Când articolul trebuie citit, el poate fi recuperat de pe disc cu întârzieri de asteptare, de căutare si rotationale mai scurte. Dacă un disc se defectează, copia este utilizată pentru serviciul cerut. Oglindirea este utilizată frecvent în aplicatii cu baze de date când accesibilitatea si viteza tranzactiilor sunt mai importante decât eficienta stocării.



Sistem de discuri "în oglindă"

RAID nivelul 2 (memorie în stilul codurilor corectoare de erori)

Sistemele de memorii asigură recuperarea din starea de disfunctie a unor componente la costuri mai reduse decât prin oglindire, dacă folosesc codurile Hamming. Codurile Hamming fac verificări de paritate pe submultimi de componente distincte si suprapuse. În una din variantele acestei scheme, patru discuri de date necesită trei discuri redundante, unul mai putin decât în cazul sistemului oglindă. Deoarece numărul de discuri redundante este proportional cu logaritmul numărului total de discuri din sistem, eficienta memorării creste odată cu numărul discurilor de date.

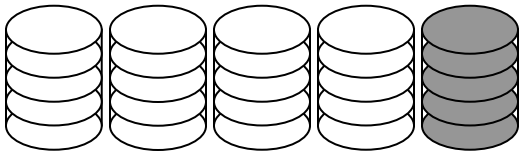


Sistem de discuri în stilul codurilor corectoare de erori

Dacă unul (si numai unul) din discuri cade, componentele mai multor discuri de paritate devin inconsistente cu datele si componenta defectă este identificată: este componenta comună tuturor subseturilor incorecte. Informatia pierdută este recuperată prin regulile obisnuite ale codului Hamming utilizat.

RAID nivelul 3 (paritate cu biti intercalati)

Se pot aduce îmbunătătiri sistemului din paragraful anterior prin observarea faptului că spre deosebire de căderea elementelor de memorie, controlerele discurilor pot identifica usor care disc este cel cu defect. Astfel, pentru recuperarea informatiei pierdute se poate utiliza un singur disc de paritate si nu un set de discuri de paritate.

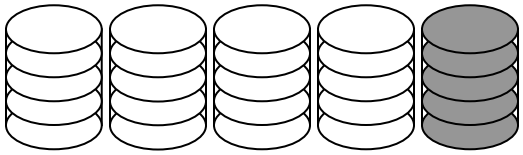


Sistem cu paritate prin biti intercalati

Într-un sistem de discuri cu paritate de biti intercalati, datele sunt conceptual intercalate bit-cu-bit pe discurile de date si se adaugă un singur disc de paritate pentru a tolera căderea unui disc (si numai a unuia). Fiecare cerere de citire accesează toate discurile de date si fiecare cerere de scriere accesează toate discurile de date si discul de paritate. Astfel numai o cerere poate fi servită la un moment dat. Deoarece discul de paritate contine numai informatia de paritate si nu date, discul de paritate nu poate participa la citiri, ceea ce produce o usoară scădere în performanta la citire fată de sistemele cu redundante care distribuie informatia de paritate si datele pe toate discurile. Sistemele de discuri cu paritate pe biti intercalati sunt utilizate frecvent în aplicatii care cer o lărgime de bandă mare dar nu viteze intrare-iesire mari. Mai sunt si simplu de implementat.

RAID nivelul 4 (paritate pe blocuri intercalate)

Există o similitudine între sistemele de discuri cu intercalare de biti si cele cu intercalare de blocuri. Deosebirea constă în obiectul operatiei de intercalare: nu intercalare de biti ci de blocuri de dimensiune arbitrară. Dimensiunea acestor blocuri este denumită unitatea de stripare (striping). Citirile cerute, mai mici decât unitatea de stripare accesează numai un disc de date. Cererile de scriere trebuie să actualizeze blocurile de date cerute si trebuie totodată să calculeze si să actualizeze blocul de paritate. Pentru scrieri de mare întindere care ating blocuri pe toate discurile, paritatea este calculată observînd cum diferă datele noi de cele vechi si aplicând acele diferente pe blocul de paritate. Scrierile de mică întindere cer asadar patru operatii de intrare-iesire pe disc: una pentru a scrie articolul nou, apoi două pentru a citi vechiul articol si vechea paritate pentru a calcula noua informatie de paritate si una de scriere a noii parităti. Această operatie este cunoscută ca o procedură citeste-modifică-scrie. Deoarece un sistem de discuri cu paritate cu intercalare de blocuri are numai un disc de paritate, care trebuie actualizat la toate operatiile de scriere, discul de paritate poate deveni cu usurintă un loc îngust, o strangulare. Din cauza acestei posibile limitări sunt de preferat sistemelor de discuri cu paritate pe blocuri, sistemele de discuri cu paritate pe blocuri distribuite.

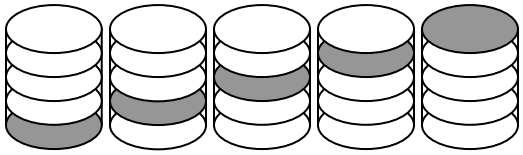


Sistem de discuri cu paritate pe blocuri intercalate

RAID nivelul 5 (paritate pe blocuri intercalate distribuite)

Sistemele de discuri cu paritate pe blocuri distribuite elimină strangularea de pe discul de paritate care se constată la sistemele cu paritate pe blocuri intercalate prin distribuirea informatiei de paritate uniform pe toate discurile. Un avantaj

suplimentar al distribuirii informatiei de paritate uniform pe toate discurile, frecvent trecut cu vederea, constă în distribuirea si a datelor pe toate discurile si nu ca în cazul anterior, pe toate cu exceptia unuia. Aceasta permite tuturor discurilor să participe la servirea operatiilor de citire spre deosebire de schemele redundante cu discuri de paritate dedicate, în care discurile de paritate nu pot participa la servirea solicitărilor de citire. Sistemele cu paritate pe blocuri intercalate distribuite au una dintre ele mai bune performante pentru citiri restrânse, citiri lungi si scrieri lungi, între toate sistemele de discuri cu redundante. Cu toate acestea, cererile de citire de lungime restrânsă sunt întrucâtva ineficiente comparativ cu schemele redundante, cum sunt cele cu oglindire, datorită necesitătii de a executa operatii citeste-modifică-scrie pentru actualizarea parităţii. Aceasta este partea slabă majoră a sistemelor RAID de nivelul 5, în ceea ce priveste performantele.



Sistem de discuri cu paritate pe blocuri intercalate distribuite

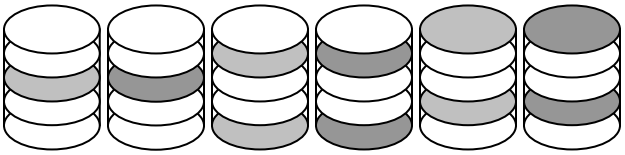
Metoda exactă utilizată pentru a distribui paritatea în sistemele cu paritate distribuită în blocuri intercalate are impact asupra performantelor. Figura alăturată ilustrează cea mai bună distributie a informatiei de paritate (discurile gri), numită distributie de paritate simetrică la stânga. O proprietate utilă a acestui gen de distributie constă în faptul că ori de câte ori sunt traversate secvential unitătile de stripare, fiecare disc este accesat în succesiune o dată înainte de a fi accesat a doua oară. Această proprietate reduce conflictele de disc atunci când sunt servite solicitările mari.

0	1	2	3	**P_0**
5	6	7	**P_1**	4
10	11	**P_2**	8	9
15	**P_3**	12	13	14
P_4	16	17	18	19

Sistem RAID de nivelul 5 cu simetrie la stânga

RAID nivelul 6 (redundante P + Q)

Paritatea este un cod redundant capabil a corecta orice defectare singulară care se autoidentifică. Dacă sunt luate în considerare sisteme cu discuri mai numeroase, este necesar a utiliza coduri mai puternice, capabile să tolereze defecte multiple. Mai mult, când un disc într-un sistem protejat prin paritate cade, recuperarea continutului discului defect reclamă lectura cu succes a continutului tuturor discurilor functionale. În aceste cazuri, probabilitatea de a întâlni în cursul recuperării o eroare de citire necorectabilă poate fi semnificativă. Asadar, aplicatiile cu cerinte de fiabilitate mai severe trebuie tratate cu coduri corectoare de erori mai puternice.



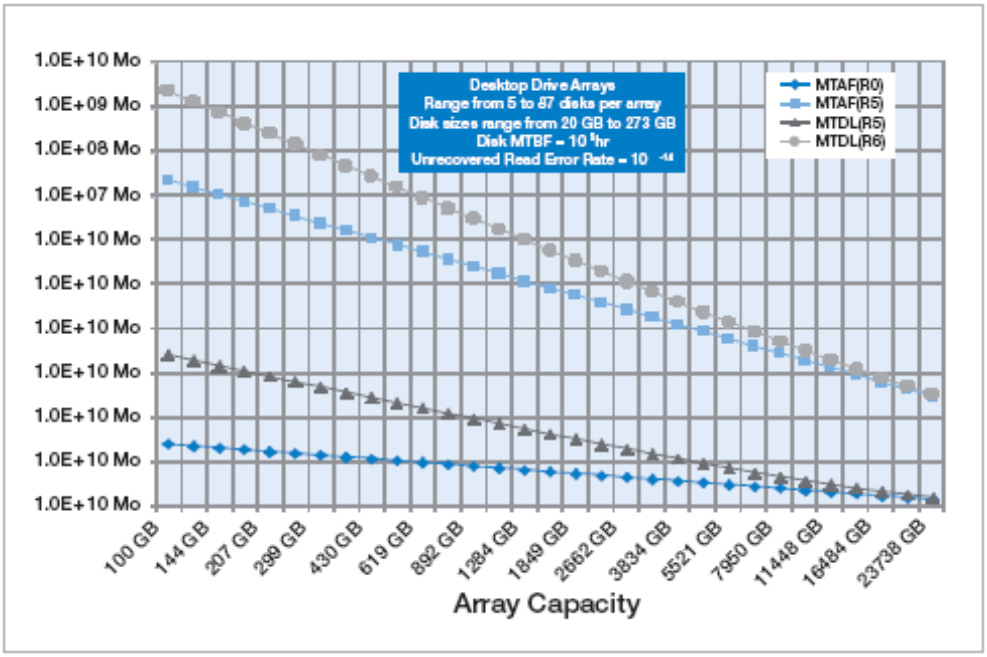
Sistem de discuri cu redundante P + Q

O astfel de schemă, denumită adesea schemă cu redundantă P + Q, foloseste codurile Reed-Solomon pentru protectia fată de căderea a până la două discuri utilizând cel putin două discuri redundante. Sistemele de discuri cu redundante P + Q sunt structural foarte asemănătoare sistemelor cu paritate distribuită bloc-intercalată si operează în mare măsură în acelasi mod. În particular, sistemele de discuri cu redundnate P + Q execută operatii scurte de scriere în maniera citeste-modifică-scrie cu deosebirea că în loc de patru accesări de disc sunt aici necesare sase accesări pentru a actualiza atât informatiile P cât si cele Q.
Sistemele RS-RAID prezentate cu mai multe detalii în prima parte a acestei sectiuni se încadrează în această clasă de sisteme de nivelul 6.

Alte sisteme RAID

Literatura mai mentionează:

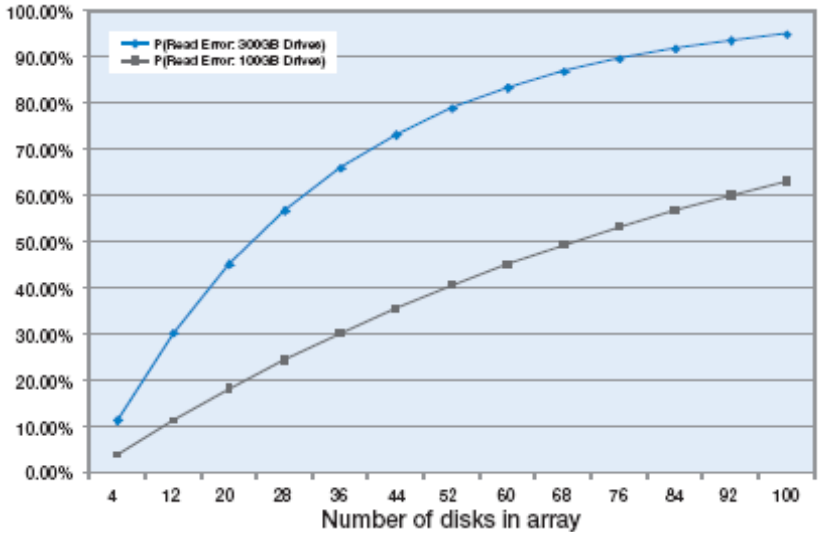
a) Sistemele de **nivel 0+1** – oglindă si stripare – cu două subsisteme 0 cu stripare si un subsistem 1 suprapus acestora. Se utilizează si pentru replicarea datelor pentru partajarea lor.
b) Sistemele de **nivel 1+0** – stripare pe oglinzi – în care sunt create subsisteme RAID 1 si peste acestea un subsistem RAID 0 de stripare.
c) Sisteme de **nivelul 7** – un sistem brevetat de Storage Computer Corporation care adaugă sectiuni cache la sistemele de nivelul 3 si 4.
d) Sisteme **RAID S** – proprietar EMC Corporation – sisteme cu paritate si stripare utilizate în sistemele proprii de memorie Symmetrix.



Timpul mediu până la defectare în functie de capacitatea de memorare

Comparatii de costuri si de performante

Primele trei măsuri prin care se evaluează un sistem de discuri sunt fiabilitatea, performanta si costul. Sistemele RAID de la 0 la 6 acoperă o gamă largă de compromisuri între aceste trei măsuri. Este important a lua în considerare toate aceste trei măsuri pentru a înțelege deplin valoarea si costul fiecărei organizări ale sistemelor de discuri. Despre fiabilitate – în graficul alăturat.
Linia cea mai de jos (romburi) reprezintă timpul mediu până la defectare pentru un singur disc. Într-un sistem fără redundante (RAID 0) aceasta produce pierderea datelor. Linia următoare (triunghiuri) arată timpul mediu până la pierderea de date, MTDL (Mean Time to Data Loss) pentru un sistem RAID 5 cu probabilitatea de a găsi un defect latent după reconstruire a informatiei. De observat că un sistem RAID 5 cu capcitatea totală mai mare de 5 TB ar putea pierde date de mai multe ori în decursul unui an. Pentru a ilustra impactul defectelor latente asupra calculului MTDL, sau timpul mediu până la eroarea aditională, MTAF (Mean Time to Additional Failure), curba următoare (pătrate) arată probabilitatea căderii din cauza a două erori de disc, care ignoră defectele latente, pentru un sistem RAID 5. Ignorarea impactului defectelor latente arată că MTDL pentru un sistem RAID 5 rămâne destul de bun chiar la capacități de peste 5 TB. Linia cea mai de sus (cercuri) arată MTDL pentru un sistem RAID 6 cu luarea în considerare a probabilității de a găsi defecte latente. Linia arată MTDL pentru sistemele RAID 6, chiar tinând seamă de impactul defectelor latente, care este cu ordine de mărime mai bun decât cel pentru un sistem RAID 5 comparabil.



Probabilitatea erorilor de citire irecuperabile în timpul reconstructiei
Rata erorilor nerecuperabile ale discului: 1 la 10^{14} biti cititi

Pentru întelegerea mai exactă a modului în care defectele latente din sistemele RAID 5 afectează MTDL să examinăm probabilitatea de a întâlni un defect latent în timpul operatiei de reconstruire. Dacă un controler de RAID 5 întâlneste un defect în timpul reconstruirii, datele utilizatorului sunt pierdute deoarece discul defect si sectorul defect reprezintă două elemente lipsă, ceea ce depăseste capacitatea sistemului RAID 5 de a recupera date pierdute. Figura alăturată arată probabilitatea de a găsi un defect latent în timpul reconstructiei sistemului la capacităti ale discurilor variate, din ce în ce mai mari. Pentru sisteme foarte mari de discuri de mare capacitate, ar fi surprinzător a nu găsi un defect latent în timpul reconstructiei. Graficul presupune o rată a erorilor tipică pentru discuri din clasa desktop. Probabilitatea este un ordin de mărime mai mică pentru discuri din clasa enterprise.

BIBLIOGRAFIE

1. Cătuneanu, V.M. si A.Mihalache *Bazele teoretice ale fiabilitatii* Ed.Academiei RSR, Bucuresti 1983
2. Chen, P.M. si altii *RAID: High-Performance, Reliable Secondary Storage* ACM Computer Surveys, 1994
3. Dumitrescu, D. si H.Costin *Retele neuronale. Teorie si aplicatii*, Ed.Teora, Bucuresti, Sibiu, 1996
4. Mihalache, A. *Când calculatoarele gresesc. Fiabilitatea sistemelor de programe (software),* Ed.Didactică si pedagogică, Bucuresti 1995
5. Plank, J.S. *A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems* plank@cs.utk.edu (1999)
6. Popovici, Al.A. *Proiectarea securităţii sistemelor complexe*, Ed.Stiintifică si enciclopedică, Bucuresti 1988
7. Stefănescu, C. *Sisteme tolerante la defecte*, Matrix Rom, Bucuresti 1999
8. Vancea, R., St.Holban si D.Ciubotariu *Recunoasterea formelor. Aplicatii*, Ed.Academiei RSR, Bucuresti 1989