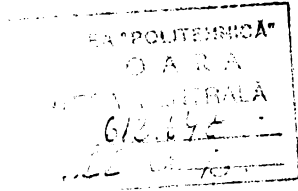


UNIVERSITATEA "POLITEHNICA" DIN TIMIȘOARA  
Facultatea de Automatizări și Calculatoare



**Ing. H. Kakas Ștefan**

**CONTRIBUȚII LA ANALIZA ȘI SINTEZA  
DE FIABILITATE  
A SISTEMELOR MULTIPROCESOR**

TEZĂ DE DOCTORAT

BIBLIOTECA CENTRALĂ  
UNIVERSITATEA "POLITEHNICA"  
TIMIȘOARA

Conducător științific:

**Prof. dr. ing. Vlăduțiu Mircea**



---

## CUPRINS

Capitolul 1. INTRODUCERE .....	5
Capitolul 2. ARHITECTURA SISTEMELOR DE CALCUL DISTRIBUITE .....	8
2.1. Clasificarea sistemelor distribuite .....	8
2.2. Sisteme cu magistrale .....	13
2.3. Rețele de interconectare statice .....	14
2.4. Rețele de interconectare dinamice .....	18
Capitolul 3. FIABILITATEA SISTEMELOR CU COMPONENTE MULTIPLE ....	25
3.1. Noțiunea de fiabilitate .....	25
3.2. Fiabilitatea în spațiul timpului continuu .....	27
3.3. Fiabilitatea în spațiul timpului discret .....	30
3.4. Distribuții fără memorarea trecutului .....	35
3.5. Funcția de structură a sistemelor bivalente .....	39
3.6. Sisteme de tip serie-parallel .....	45
3.7. Sisteme de tip multiterminal .....	53
3.8. Sisteme duale; căi și tăieturi .....	64
3.9. Utilizarea tehnicii booleene pentru calculul fiabilității .....	81
3.10. Disjunctizarea termenilor din funcția de structură .....	85
3.11. Evaluarea fiabilității unei rețele cu relații logice de semnal .....	104
3.12. Importanța legăturilor în contextul fiabilității rețelei .....	112
3.13. Sisteme cu defecțiuni dependente .....	116
3.14. Fiabilitatea sistemelor cu stări multiple .....	119
Capitolul 4. FIABILITATEA SISTEMELOR MULTIPROCESOR .....	124
4.1. Caracteristici generale de fiabilitate și performanță .....	124
4.2. Modelul integrat de fiabilitate și performanță propus .....	137
4.3. Rezultate experimentale .....	151
Capitolul 5. FIABILITATEA REȚELELOR DE CALCULATOARE .....	165
5.1. Caracteristici generale de fiabilitate și performabilitate .....	165
5.2. Modelul de performabilitate propus .....	187
5.3. Validarea modelului .....	194

Capitolul 6. CONCLUZII .....	198
ANEXA 1	
Determinarea termenilor disjuncți pentru calculul fiabilității pe baza vectorilor de cale .....	202
ANEXA 2	
Determinarea expresiei de fiabilitate prin minimizarea disjunctă a funcției de structură booleene .....	206
ANEXA 3	
Ordonarea legăturilor după importanța lor în fiabilitatea $s-t$ .....	208
ANEXA 4	
Algoritmii de simulare a comportării sistemelor multiprocesor .....	210
BIBLIOGRAFIE .....	215

## 1. INTRODUCERE

Progresele tehnologice în fabricația circuitelor VLSI au dus la apariția unor circuite integrate capabile să conducă procese parțiale relativ complexe ale unui sistem de calcul. Acest lucru a determinat realizarea sistemelor distribuite, care pe lângă o valoare mai ridicată a raportului performanță/cost ca urmare a activităților paralele, asigură și un înalt grad de modularitate, o divizare eficientă a resurselor și permit toleranța superioară a defectelor. La proiectarea unor astfel de sisteme un loc important îl ocupă evaluarea fiabilității acestora, pentru care s-au dezvoltat diverse principii și metode de calcul prezentate în literatura de specialitate. Oportunitatea alegerii temei a fost dată de răspândirea masivă a sistemelor cu mai multe procesoare cu diferite grade de cuplaj între ele. În context fiabilistic, asemenea sisteme trebuie privite pe de o parte ca fiind incluse în categoria sistemelor cu componente multiple, iar pe de altă parte ca fiind predispuse însăși prin structura lor (cu multiple unități de calcul, legături de comunicație, fișiere de programe și date) la proprietăți de toleranță la defecte. Pentru inginerul de fiabilitate, sistemul multiprocesor strâns cuplat sau slab cuplat se prezintă ca o colecție de resurse (procesoare, memorii, periferice, baze de date etc.) interconectate printr-o rețea de comunicație și controlate de un sistem de operare distribuit. Modul de interconectare a resurselor prin liniile de comunicație și deci căile posibile pentru transferul datelor între procesoare determină topologia sistemului. Există diverse clasificări ale topologiilor, deosebindu-se topologii de tip static sau dinamic, orientate pe legături sau pe magistrale etc. Alegerea unei anumite topologii depinde de o mulțime de factori cum ar fi expandabilitatea, flexibilitatea, toleranța la defecte, strategia de implementare a algoritmilor și nu în ultimul rând fiabilitatea.

Fiabilitatea se definește matematic ca fiind probabilitatea pentru ca un produs să-și îndeplinească funcțiile prevăzute în condiții prestabilite pe o durată dată, presupunând că a fost funcțional la momentul inițial. Pentru un sistem cu mai multe procesoare, fiabilitatea se traduce în general în existența a cel puțin unei căi de comunicație între oricare pereche de procesoare. Modelele de fiabilitate uzuale încearcă să dea o imagine cât mai realistă a comportării sistemului, dar conțin de regulă clauze simplificatoare datorită calculului foarte complex impus de exemplu de numărul foarte mare de stări. Există astăzi mulți algoritmi performanți de calcul, dar problema nu este rezolvată integral fiind neglijată uneori factori cu influență asupra fiabilității. De exemplu, într-o rețea de calculatoare desființarea conectivității totale poate afecta cel mult un procent redus de utilizatori. De aceea, în asemenea sisteme eficacitatea sau nivelul de performanță al sistemului reprezintă măriri mai realiste de evaluare a fiabilității, problematica performabilității fiind de maximă actualitate.

Această teză își propune în principal o abordare constructivă a problemelor de analiză și evaluare a fiabilității sistemelor cu procesoare multiple pornind de la câteva arhitecturi de bază, precum și dezvoltarea de modele originale unificate de performanță și fiabilitate pentru acestea. În această teză nu am admis și posibilitatea reparației elementelor defecte, deci nu tratez aspectul disponibilității sistemelor. Lucrarea se întinde pe șase capitole, urmate de patru anexe și o listă bibliografică.

După un scurt capitol de introducere, în capitolul 2 fac o delimitare a tipurilor de sisteme multiprocesor sub aspect arhitectural, precizând în fiecare caz metodele de

efectuarea a transferurilor de date și caracteristicile de trafic. Definierea structurilor reprezentative de sisteme cu mai multe procesoare va fi folosită ca punct de referință în capitolele următoare la construirea modelelor de performabilitate pentru aceste sisteme.

În capitolul 3 introduc și dezvolt aparatul matematic necesar pentru calculul fiabilității sistemelor cu componente multiple atât în spațiul continuu, cât și, ca o contribuție personală utilizată apoi în capitolul următor, în spațiul discret al timpului. Structurată în principiu pe definiții, leme și teoreme (o parte originale), autorul demonstrează pe numeroase exemple utilitatea practică a soluțiilor propuse. Accentul principal se pune îndeosebi pe calculul fiabilității sistemelor coerente reprezentate prin grafuri, care este în general o problemă NP-hard. Tocmai de aceea literatura actuală de specialitate îi acordă un spațiu foarte larg, încercând să găsească, măcar pentru anumite cazuri particulare, simplificări și algoritmi de calcul în timp polinomial. Într-o rețea modelată printr-un graf probabilistic, problema fiabilității de tip  $K$ -terminal constă în determinarea probabilității ca o submulțime  $K$  de noduri să poată comunica între ele. În general metodele prezentate se axează pe două direcții principale: reducția grafului cu conservarea fiabilității și calcule analitice pe baza funcției de structură, care poate fi scrisă conform dualității atât pe baza mulțimii căilor minimale, cât și a tăieturilor minimale. La ambele tehnici aduc contribuții proprii în sensul extinderii aplicabilității, respectiv al unor simplificări în determinarea expresiei de fiabilitate. Metodele analitice se bazează pe utilizarea relațiilor de calcul probabilistic, fiind în esență de patru feluri: enumerarea stărilor, principiul de incluziune-excluziune, suma de produse disjuncte și descompunerea pivotală. Avantajele și dezavantajele fiecărei metode au fost desprinse în urma unui studiu comparativ al efortului de calcul și preciziei. În contrast cu efectuarea de calcule probabilistice, uneori greu de stăpânit ca și complexitate, transpunerea problemei în spațiul boolean oferă posibilitatea exploatării puternicelor teoreme ale algebrei booleene. Forma cea mai simplă a expresiei de fiabilitate se obține prin luarea în considerare a cuburilor disjuncte de pe diagrama de minimizare aferentă funcției de structură, ceea ce face ca metoda să fie clară și comodă de utilizat. În acest context disjunctizarea termenilor din funcția de structură, însoțită de o preprocesare a acestora, capătă valențe majore în literatura de specialitate. Deoarece reprezentarea pe diagrama Karnaugh nu permite tratarea sistemelor cu mai mult de șase componente, procedee de genul descompunerii de tip conservativ sau exhaustiv a funcției booleene de structură, respectiv a decompoziției acesteia, adoptată ca și o contribuție personală, par a fi inevitabile în cazul rețelelor uzuale. În multe situații practice, complexitatea sistemului și numărul mare de componente nu permit o determinare exactă a fiabilității. De aceea am considerat utilă o analiză a principalelor metode de aproximare în calculul indicatorilor de fiabilitate. Pe lângă metodele convenționale de calcul a fiabilității de tip  $K$ -terminal dintr-o rețea, prezentarea cuprinde și alte procedee, cum ar fi cea bazată pe căile minimale de bază, metoda holografică sau cea a relațiilor logice de semnal. De asemenea sunt baleiate sumar modalitățile de ordonare a legăturilor după aportul adus la valoarea indicatorului de fiabilitate, aspect semnificativ în sinteza de rețele fiabile utilizând componente mai puțin fiabile. În practică se întâlnesc frecvent situații în care o evaluare mai realistă a fiabilității unui sistem cu multe componente se obține prin luarea în considerare a dependenței dintre defectări, respectiv a multitudinii stărilor sistemului și componentelor. În scopul de a putea constitui o bază pentru cercetări viitoare, am inclus și o scurtă prezentare a tendințelor actuale în aceste direcții.

După trecerea în revistă a principalelor criterii de fiabilitate și performanță a sistemelor multiprocesor strâns cuplate, în capitolul 4 construiesc un model integrat de performanță și fiabilitate, care ține cont și de posibilitatea defectării elementelor de legătură, de regulă cele mai puțin fiabile din cadrul unui astfel de sistem. Modelul, definit

într-un spațiu discret al timpului adecvat pentru operațiile de transfer efectuate în cadrul unui singur ciclu instrucțiune sau ciclu mașină, este aplicat succesiv la principalele sisteme multiprocesor din această categorie: crossbar, cu magistrale și cu rețea de interconectare multinivel. Compararea rezultatelor furnizate de modelul matematic cu cele experimentale obținute prin simulare cu calculatorul, au confirmat în toate cazurile valabilitatea supozițiilor teoretice. Acest lucru a permis utilizarea modelului pentru formularea unor concluzii comparative utile în activitatea de analiză și a unor recomandări de sinteză optimă sub aspectul raportului performanță/cost a sistemelor de acest gen.

Capitolul 5 se ocupă de fiabilitatea sistemelor multiprocesor slab cuplate, în speță a rețelelor de calculatoare. Evidențierea caracteristicilor de fiabilitate a acestor sisteme se poate face printr-o investigare pe bază de criterii deterministe, criterii probabiliste sau urmărind distribuția programelor și fișierelor în rețea. Dacă metodele din prima categorie țin cont exclusiv de topologia rețelei folosind tehnici din teoria grafurilor, în al doilea caz se folosește un model stohastic de rețea prin asocierea de probabilități de defectare legăturilor și nodurilor din graful asociat. Acest model servește pentru o evaluare cantitativă a indicatorilor de fiabilitate de tip terminal și multiterminal, acțiune necesară în faza de analiză preventivă a unei rețele. Fiabilitatea legată de execuția programelor distribuite în rețea ia în considerare în plus și modul în care s-a făcut alocarea copiilor multiple ale programelor și fișierelor de date la procesoarele aflate în nodurile rețelei. O direcție de cercetare care aduce mai aproape de realitate studiul fiabilității, în special la sisteme tolerante la defecte cum sunt și rețelele de calculatoare, este performabilitatea. În acest caz se îmbină caracteristicile legate de defectarea componentelor cu performanțele de exploatare sesizate la nivelul sistemului. La modelul de performabilitate propus pentru o rețea de calculatoare am ținut cont de încărcarea cu trafic a liniilor de transmisie dintre noduri, adoptând un formalism matematic în timp continuu. Indicatorul de performabilitate introdus, accesibilitatea instantanee, urmărește existența unei căi operaționale libere între perechea de noduri observate. Validarea modelului care calculează intensitatea traficului pe fiecare linie s-a făcut prin comparare cu rezultatele date de un program de simulare a emiterii și dirijării pachetelor de date printr-o rețea de referință.

Ultimul capitol conține concluziile și o sinteză a contribuțiilor originale aduse de autor pe parcursul tezei, evidențind și unele direcții posibile de cercetare.

Autorul dorește să mulțumească și pe această cale conducătorului științific, domnul prof.dr.ing.Vlăduțiu Mircea, pentru îndrumarea deosebit de competentă și de înaltă ținută profesională care i-a fost acordată de-a lungul întregii perioade de pregătire, îndreptată cu exigență și tact în scopul finalizării eficiente a tezei.

## 2. ARHITECTURA SISTEMELOR DE CALCUL DISTRIBUITE

### 2.1. Clasificarea sistemelor distribuite

Sistemele distribuite acoperă un spectru larg în cadrul sistemelor de calcul și formularea unei definiții exacte este dificilă. Un calculator distribuit constă dintr-o mulțime de procesoare, având fiecare la un moment dat câte un spațiu distinct de adrese și care sunt interconectate printr-un subsistem de comunicație [LEVI94]. Un asemenea calculator folosește conceptul de prelucrare paralelă (parallel processing) ca o metodă de organizare a operațiilor în care mai mult de o operație este realizată în mod concurrent. Spre deosebire de un program secvențial, o operație începută nu presupune totdeauna desăvârșirea celei anterioare. Două caracteristici sunt însă relevante în acest caz, și anume sincronizarea între evenimentele corelate și comunicarea între procesele concurente. *Sistemul distribuit* (distributed system) poate fi definit ca un sistem care conține un program distribuit care rulează pe un calculator distribuit. S-a introdus și noțiunea de sistem complet distribuit, în care un procesor nu poate primi la un moment dat o imagine coerentă asupra întregului sistem [NEME93]. Pe baza definiției, un sistem în care un program distribuit rulează pe un singur procesor sau în care un program secvențial rulează pe un calculator distribuit, nu poate fi considerat un sistem distribuit.

Criteriul cel mai uzual de clasificare a sistemelor distribuite este după *granularitatea* (granularity) interacțiunilor dintre activitățile care sunt executate în paralel. Fie un sistem cu  $n$  procesoare, în care o parte a timpului de lucru al fiecărui procesor este afectată schimburilor de date și informații de control cu alte procesoare. Instrucțiunile executate vor fi deci sau instrucțiuni de calcul, sau instrucțiuni de comunicare. Presupunem că rezolvarea unei anumite probleme consumă un timp total de prelucrare  $T$ , dat de suma timpilor necesari pentru fiecare procesor. Notând cu  $t_i^c$  timpul petrecut de procesorul  $i$  pentru comunicare și cu  $t_i^w$  restul timpului afectat pentru calcul, timpul total de comunicare pentru toate procesoarele va fi

$$T^c = \sum_{i=1}^n t_i^c \quad (2.1.1)$$

iar timpul total de calcul pentru toate procesoarele:

$$T^w = \sum_{i=1}^n t_i^w \quad (2.1.2)$$

Se observă că nu s-a făcut nici o presupunere privind repartizarea uniformă a timpului de prelucrare la fiecare procesor. Timpul total de prelucrare va fi dat de:

$$T = T^c + T^w \quad (2.1.3)$$



Cu aceste notații, **granularitatea**, numită și **granulația de prelucrare** (grain of computation) sau **balanța prelucrării** (computation balance), reprezintă raportul dintre timpul afectat pentru calcul și cel afectat pentru comunicare, adică:

$$\gamma = \frac{T^w}{T^c} \quad (2.1.4)$$

Este important de remarcat faptul că noțiunea de granularitate exprimă caracterul reunit al unui program paralel, al arhitecturii pe care se execută și al dimensiunii problemei de rezolvat. Astfel se poate produce o modificare a granularității dacă problema de rezolvat presupune alte caracteristici de comunicație sau dimensiunile sale variază raportat la numărul de procesoare utilizate [KRUS88]. Pentru a măsura această deplasare a granularității, pe lângă granularitatea potențială (potential granularity) dată mai sus, s-a introdus încă o definiție, cea a granularității actuale (actual granularity). Aceasta consideră granularitatea ca fiind media timpului dintre două comunicări succesive:

$$\gamma' = \frac{T^w}{S} \quad (2.1.5)$$

În relația (2.1.5)  $S$  reprezintă numărul total de comutări de la calcul la comunicare pentru toate procesoarele, adică:

$$S = \sum_{i=1}^n s_i \quad (2.1.6)$$

unde  $s_i$  indică de câte ori a apărut o astfel de comutare la procesorul  $i$ . Profilul de activitate al acestuia este o secvență de timpi de calcul ( $t_{i,j}^w$ ) și timpi de comunicare ( $t_{i,j}^c$ ) de forma

$$t_{i,1}^w, t_{i,1}^c, t_{i,2}^w, t_{i,2}^c, \dots, t_{i,q}^w, t_{i,q}^c$$

astfel încât  $\sum_{j=1}^q t_{i,j}^w = t_i^w$  și  $\sum_{j=1}^q t_{i,j}^c = t_i^c$ . În general procesarea pe un sistem distribuit se poate situa între două extreme: durate scurte de calcul, urmate de durate scurte de comunicare sau durate lungi de calcul, urmate de durate lungi de comunicare. În primul caz execuția are o granularitate fină (fine granularity), iar în al doilea o granularitate grosolană (coarse granularity). Cu toate că granularitatea depinde atât de algoritm, dimensiunea problemei, cât și de arhitectură, unele sisteme de calcul favorizează un anumit tip de granulație de prelucrare. Astfel de exemplu un sistem cu granulație fină, capabil să comunice rapid date singulare este relativ costisitor, deoarece necesită un sistem de comunicație foarte complex care să evite așteptările cauzate de conflicte. Dacă  $\gamma$  este mic, fiecare procesor execută puține calcule între comunicări, fiind adecvat un sistem cu granulație fină. Dacă  $\gamma'$  este mare, atunci comunicările între procesoare sunt rare, prelucrarea fiind eficientă pe un sistem cu granulație grosolană. În concluzie, pe măsura reducerii granularității, cresc costurile de comunicație.

Având în vedere caracteristica de granularitate, sistemele distribuite se împart în două categorii mari:

- sisteme stâns cuplate
- sisteme slab cuplate

Subliniez aici că încadrarea unui sistem real într-o anumită clasă, precum și delimitarea clară a acestora este foarte dificilă și câteodată controversată în literatura de specialitate, de aceea cele prezentate trebuie considerate a fi valabile sub rezerva termenului "în general". *Sistemele strâns cuplate* (tightly/closely coupled systems), la care se consideră că  $\gamma < 10$ , constau dintr-un număr de elemente de procesare PE (processing element) și module de memorie comună CM (common memory), având între ele o rețea de interconectare (interconnection network) (figura 2.1.1). Această structură nu exclude și existența unor memorii locale LM (local memory) pentru procesoare, alături de resursele I/O. Datorită faptului că modulele CM sunt adresabile global de către fiecare PE, servind astfel (și) pentru comunicarea dintre procesoare, aceste arhitecturi se mai numesc sisteme multiprocesor cu memorie divizată (shared memory multiprocessor). Pe de altă parte, *sistemele slab cuplate* (loosely coupled systems), la care  $\gamma > 10$ , constau din procesoare cu câte o memorie locală, fără resurse comune, conectate între ele printr-o rețea de interconectare numită și rețea de comunicare (communication network) (figura 2.1.2). Tabak consideră că diferența esențială între cele două tipuri de sisteme este cea referitoare la conceptul de autonomie formulat de Enslow [TABAK90]. Astfel un sistem strâns cuplat dispune de regula de un control centralizat, pe când unul slab cuplat se conformează principiului de autonomie cooperativă. Pe baza acestui principiu, care caracterizează funcționarea și interacțiunile dintre resurse (logice și fizice), se permite unei resurse refuzul de a servi o cerere dacă este ocupată cu o altă sarcină, fără a exista o ierarhie a controlului în sistem. Németh consideră că sistemele slab cuplate sunt sisteme multiprocesor la care transferul dintre două procesoare se realizează astfel încât unul îl privește pe celălalt ca un echipament periferic [NÉME93]. Uneori sistemele slab cuplate sunt considerate și rețele de calculatoare (computer networks), deși în [RAI90a] se includ în această categorie numai rețelele locale LAN (local area network). Pentru rețelele WAN (wide area network) se introduce o altă clasă de sisteme distribuite, și anume sistemele abia cuplate (barely coupled systems), care au  $\gamma \approx 100$ .

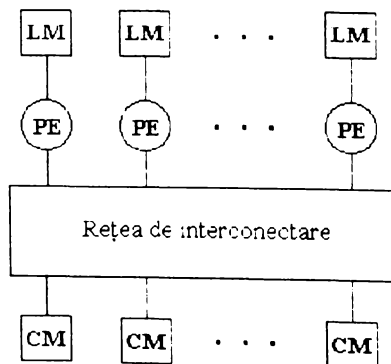


Figura 2.1.1.

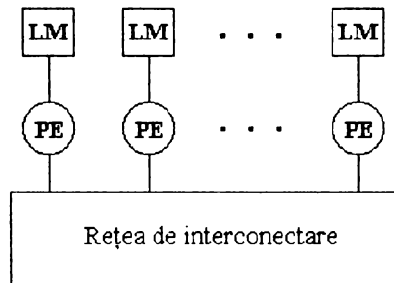


Figura 2.1.2.

În literatura de specialitate, prin termenul de *sistem multiprocesor* (multiprocessor system) se subînțelege de regulă un sistem de tip MIMD (multiple-instruction multiple-data stream), cu mai multe șiruri de instrucțiuni și mai multe șiruri de date, conform clasificării lui Flynn [TABAK90][WILK91]. Acest sistem este diferit de unul M/SISD (multiple/single-instruction single-data stream), care presupune independența unităților de control și de execuție, atât sistemele M/SISD, cât și cele SISD fiind cazuri degenerate ale lui MIMD. Într-un sens mai larg, un sistem multiprocesor reprezintă o structură de calcul în care are loc execuția concurrentă a sarcinilor separate aparținând aceleiași algoritm. Un sistem multiprocesor se consideră a fi fiabil pentru multiprocesare (multiprocessing reliable) dacă și numai dacă poate fi utilizat atât pentru operații MIMD, cât și M/SISD [HWANG82]. Sistemele SIMD (single-instruction multiple-data stream), la care mai multe procesoare execută simultan aceeași instrucțiune asupra datelor diferite sunt implementate sub forma arilor de calculatoare (computer array), cele MISD (multiple-instruction single-data) neavând un corespondent practic [WILK91]. În conceptul general al sistemelor paralele se încadrează și mașinile paralele speciale [AJMO86] sau arhitecturile neconvenționale (de tip bandă de asamblare, flux de date, arhitecturi sistolice, calculatoare vectoriale), dar care nu fac obiectul acestei prezentări.

În ceea ce privește tipul de comunicare între elementele componente ale sistemului prin rețeaua de interconectare, aceasta poate fi în principiu de două tipuri:

- prin comutare de circuite
- prin comutare de mesaje

În cazul *comutării de circuite* (circuit switching) trebuie stabilită o cale directă de comunicație între cele două părți implicate, înainte de începerea transferului. Aceasta presupune existența unei legături directe între sursă și destinație, permițând o viteză mare de transfer întrucât după înfăptuirea legăturii nu este necesară adăugarea unor informații de adresare la datele transferate. Existența unei legături permanente nu implică însă și o cale fizică continuă între cele două unități, acest lucru fiind evident de exemplu într-un sistem cu magistrală comună utilizată de procesoare prin tehnici de divizare a timpului sau multiplexare. *Comutarea de mesaje* (message switching) este o metodă de transmitere a datelor în care acestea se transferă prin puncte intermediare (noduri) de pe calea de comunicație, în care sunt reținute pentru scurt timp și apoi retransmise spre destinație. Nodurile prin care se trimite mai departe informația pot fi procesoare și/sau comutatoare, iar din mesaj face parte și o adresă care servește pentru direcționarea acestuia înspre destinație. O rețea cu comutare de mesaje poate deveni foarte lentă dacă mesajele sunt

lungi, deoarece acestea vor "bloca" rețeaua față de celelalte procesoare. Mesajele foarte scurte nu sunt însă eficiente datorită informațiilor de adresare și corectoare de eroare, care sunt suplimentare față de cele de date. Există și comunicația prin *comutare de pachete* (packet switching), similară cu cea precedentă, cu deosebirea că la sursă mesajul de transmis este desfăcut în mai multe pachete individuale care se trimit prin rețea, urmând ca la destinație să fie reasamblate pentru a forma mesajul original. Operarea cu succes a unei rețele în acest caz depinde foarte mult de existența unui algoritm de dirijare corespunzător și fiecare nod trebuie să aibă incorporat un control de dirijare (routing control) pentru a direcționa corect mesajele prin rețea. Această operație necesită însă un anumit timp, de aceea este preferabilă amplasarea cât mai aproape a procesoarelor care comunică frecvent. Rețelele cu comutare de mesaje/pachete oferă câteva avantaje față de cele cu comutare de circuite [ANDE89]:

- o conectare rapidă, datele de transmis fiind de regulă acceptate imediat
- rată scăzută a erorilor, asigurată prin atașarea informațiilor de verificare
- fiabilitate mai mare, datorită existenței de căi alternative
- sincronizare între elementele care schimbă date

Arhitecturile sistemelor distribuite pot fi analizate și sub alte aspecte, cum ar fi expandabilitatea, capacitatea și flexibilitatea [RAI90a]. Expandabilitatea sistemului este o măsură a abilității sale de a prelua sarcini mărite, astfel încât pentru o nouă aplicație să nu fie necesară nici reproiectarea completă a hardware-ului și nici a software-ului de comunicație. Capacitatea exprimă proprietatea rețelei de a putea mări cantitatea de date transferate atunci când se adaugă noi procesoare la sistem. Flexibilitatea se referă la posibilitatea de adaptare a sistemului la o modificare a fluxului de date, de exemplu atunci când este necesară redirijarea unui mesaj din anumite cauze (procesor ocupat sau defect). De asemenea, sistemele distribuite sunt adecvate pentru implementarea *toleranței la defecte* (fault tolerance), datorită resurselor hardware multiple din componență. Comportarea rezistentă la defecte (fault resilient) decurge în general din conectivitatea nodurilor și muchiilor grafului asociat sistemului. Dacă sistemul poate furniza corect informația cerută de aplicație chiar cu unele legături/noduri defecte, acestea vor fi tolerate de sistem fără efecte negative asupra funcționării. În multe situații toleranța este posibilă doar cu reducerea performanțelor până la o limită admisibilă, adică are loc o degradare "grațioasă" (graceful degradation). În primul caz se mai spune că sistemul are o comportare de tip "fail-operational", iar în al doilea de tip "fail-soft" [GÖRKE89]. Într-un sistem distribuit, diagnoza permite localizarea defectelor astfel încât sistemul să poată fi reconfigurat în jurul unităților rămase operaționale. Degradarea grațioasă este înfăptuită printr-o reconfigurare automată și capacitate de recuperare (recovery) furnizate cu ajutorul unei mentenanțe on-line [DAS87]. Aceste sisteme dispun de asemenea de un mediu în care testele software și diagnoza fac parte integrantă din încărcarea normală a sistemului [PRAD86].

Sistemele multiprocesor conțin o mulțime de elemente procesoare care, deși individual sunt complet programabile și capabile să execute propriile programe, formează o singură entitate, fiind coordonate de un singur mecanism (centralizat sau distribuit) către un țel comun. În acest caz schimburile de date între elementele din sistem sunt relativ frecvente și de aceea rețeaua de interconectare are o influență majoră asupra performanțelor sistemului. După tipul acestei rețele, sistemele multiprocesor se pot clasifica în sisteme cu magistrală, cu rețea de interconectare statică și cu rețea de interconectare dinamică.

## 2.2. Sisteme cu magistrale

Un sistem multiprocesor orientat pe magistrală (bus-oriented system) conține una sau mai multe magistrale (care includ liniile de date, adrese și control) prin care se interconectează toate modulele din sistem. Cel mai simplu și ieftin mod de implementare îl reprezintă sistemul cu *magistrală unică* (single-bus), care se utilizează prin divizarea timpului de către toate elementele de procesare (PE) care doresc acces la memoria comună (CM) (figura 2.2.1). Un astfel de sistem oferă o mare flexibilitate de configurare, dar prezintă dezavantajul strângulării de magistrală (bus bottleneck) prin faptul că la un moment dat numai un PE poate efectua un transfer de date pe magistrală. Întrucât alte procesoare care simultan doresc accesul la magistrală trebuie să intre în așteptare, se reduce considerabil debitul global (overall throughput) al sistemului multiprocesor. Prioritatea între module se stabilește prin utilizarea unei scheme de arbitraj paralelă sau serială [WILK91]. Traficul pe magistrala sistem se reduce dacă fiecare procesor are câte o memorie locală (LM), care se conectează la acesta alături de resursele I/O printr-o magistrală locală, formându-se astfel un sistem ierarhic de magistrale (hierarchical bus system) [HWANG93]. Un alt dezavantaj al sistemului cu magistrală unică îl reprezintă faptul că defectarea magistralei constituie o defecțiune catastrofală echivalentă cu căderea sistemului. Aceste dezavantaje pot fi în mare măsură eliminate prin utilizarea unui sistem cu *magistrale multiple* (multiple buses) (figura 2.2.2). Procesoarele pot folosi oricare magistrală liberă pentru un acces la o memorie comună, dar simultan nu pot fi îndeplinite decât până la  $b$  cereri de transfer, dacă  $b$  este numărul de magistrale. Și aici este necesară o logică de arbitraj pentru a rezolva cererile simultane, astfel încât prima dată se selectează nu mai mult de o cerere pentru fiecare modul CM, iar apoi din acestea se selectează maximum  $b$  cărora li se alocă câte o magistrală. Numărul de procesoare care așteaptă datorită conflictelor de magistrală se reduce pe măsură ce numărul de magistrale este mai mare, dar prin aceasta crește și complexitatea sistemului. Aceste structuri se recomandă cu precădere în sistemele tolerante la defecte. Apare însă aici o altă problemă, și anume aceea ca modulele din sistem să ofere facilitatea de conectare multiport (multiporting) la magistralele sistem. Aceasta necesită câte o logică suplimentară la fiecare modul, complexitatea și costul căreia nu permite creșterea excesivă a lui  $b$  ( $b=2$  sau 3 sunt valori uzuale). Utilizarea unei memorii comune de tip multiport (multiport memory), cu linii separate pentru fiecare procesor, reprezintă în sine o arhitectură posibilă pentru un sistem multiprocesor, dar scumpă dacă  $n$  este mare (figura 2.2.3). Ca urmare se preferă soluția cu memorie "pseudomultiport", care în aparență permite accesul simultan la mai mult de o locație de memorie, dar de fapt accesele sunt secvențiale însă la o viteză ridicată [WILK91].

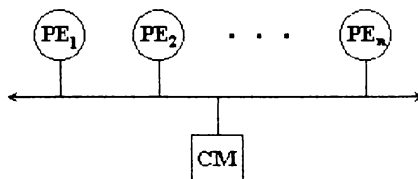


Figura 2.2.1.

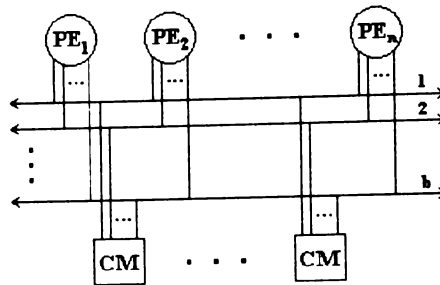


Figura 2.2.2.

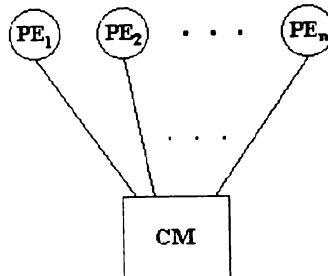


Figura 2.2.3.

### 2.3. Rețele de interconectare statice

*Rețelele de interconectare statice* (static interconnection networks) sunt acelea care permit numai legături fixe (unidirecționale sau bidirecționale) între două elemente de procesare (noduri). Acestea sunt potrivite pentru conexiuni de tip procesor-procesor, adică toate nodurile sunt procesoare și acestea preiau datele sosite pentru a le prelucra sau a le transmite mai departe la alte procesoare. Procesoarele dispun numai de câte o memorie locală, fără să mai existe o memorie comună în sistem. Dacă apare necesitatea ca un procesor să transfere date la un alt procesor, atunci acest lucru se face prin transmisie de mesaje. Deoarece fiecare nod conține de fapt câte un calculator, aceste tipuri de multiprocesoare se mai numesc multicalculatoare cu transmitere de mesaje (message-passing multicomputers) [WILK91].

Exemple de topologii de interconectare statice sunt date în figura 2.3.1. Rețeaua de tip *magistrală globală* (global bus) limitează numărul de noduri ( $n$ ), deoarece densitatea mesajelor crește liniar cu  $n$ . Conexiunile procesoarelor într-o *arie liniară* (liniar array) sunt limitate la cei doi vecini apropiați, iar prin legarea capetelor libere între ele se formează o structură în *inel* (ring). Aceasta este recomandabilă numai în sisteme cu puține procesoare, deoarece întârzierea medie de transmitere a mesajelor și densitatea traficului cresc liniar cu  $n$ . Întârzierea medie corespunde la  $n/2$  legături, iar densitatea medie a mesajelor în fiecare din cele  $n$  legături este de  $n/2$  mesaje pe unitatea de timp (presupunând că într-o unitate de timp fiecare nod inițiază un mesaj) [WITT81]. Rețelele

în inel nu au fiabilitate ridicată, deoarece defectarea unui singur nod sau a unei singure legături întrerupe cele mai multe comunicări, mesajele fiind circulate într-un singur sens.

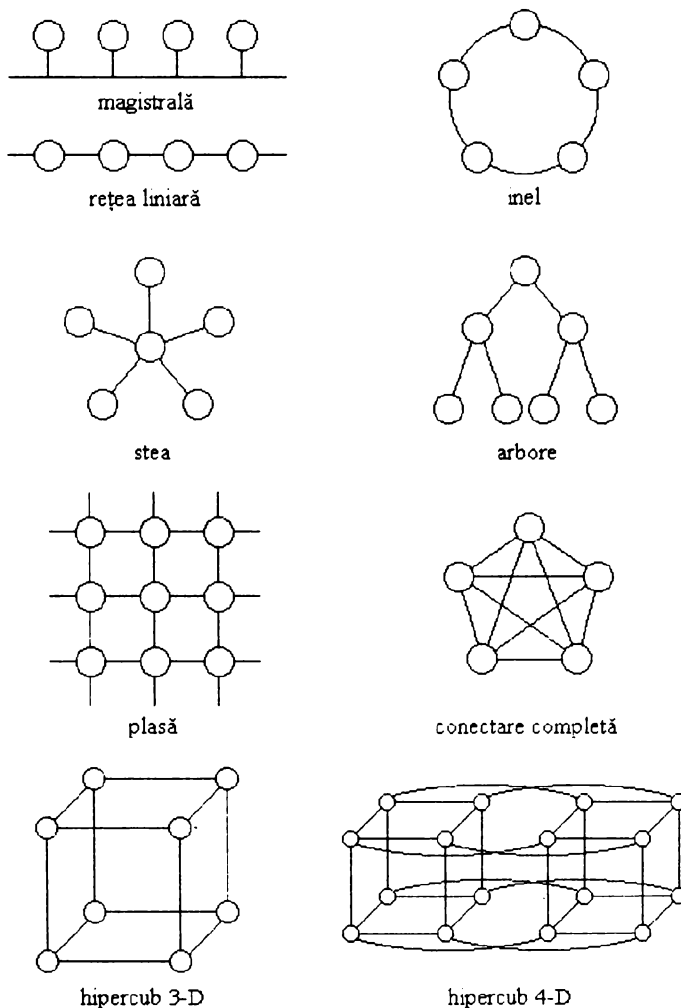


Figura 2.3.1.

Conexiunea de tip *stea* (star) este adecvată pentru  $n$  mic, când controlul rețelei din nodul central este relativ simplu. Deoarece orice pereche de noduri laterale poate comunica prin numai două legături, întârzierea introdusă este mică. Fiecare nod lateral necesită numai o conexiune, pe când cel central are  $n-1$ , ceea ce este un factor de limitare pentru  $n$ . Nodul central trebuie să transmită mai departe toate datele către destinație, ceea ce face din acesta un element de strângere a traficului din rețea, care limitează de

asemenea dimensiunea rețelei. Dacă fiecare din cele  $n-1$  noduri laterale are de transmis și de recepționat un mesaj, atunci fiecare legătură este utilizată de două ori și nodul central trebuie să facă față la un trafic de  $n-1$  mesaje. O defecțiune la nodul central (hub), care este elementul cel mai scump, poate duce la căderea întregului sistem. Costurile liniilor (în total  $n-1$ ) și conexiunilor sunt însă mici și cresc liniar cu  $n$ .

Rețelele în formă de *arbore* (tree) sunt oarecum similare cu cele de tip stea privind transmiterea mesajului între două noduri, în sensul că nodul rădăcină se folosește pentru intermedierea transferului de la o ramură a arborelui la cealaltă. În general de la nodul sursă se parcurge arborele în sus până când se întâlnește un nod comun cu calea spre nodul destinație, după care se parcurge această cale în jos. Numărul de legături ale rețelei este de  $n-1$ , ceea ce înseamnă o creștere liniară a costurilor cu dimensiunea. Defectarea unui nod critic (cel rădăcină sau apropiat de el) are urmări catastrofale asupra întregii rețele. Într-o rețea regulată cu  $b$  ramuri sub fiecare nod și cu  $j$  niveluri, numărul total de noduri este:

$$n = N(j) = 1 + b + b^2 + \dots + b^{j-1} = \frac{b^j - 1}{b - 1} \quad (2.3.1)$$

Pentru a calcula densitatea traficului de mesaje pe o legătură de deasupra unui nod din nivel  $k$ , se consideră un timp de  $n-1$  unități în care în medie fiecare nod va trimite câte un mesaj la celălalt. În acest timp fiecare nod din subarborele de sub nodul considerat (care are  $j-(k-1)$  niveluri) va schimba o pereche de mesaje între el și restul de noduri ale rețelei. Ca urmare densitatea  $T(k)$  a traficului pe linia de deasupra nodului de nivel  $k$  va fi:

$$T(k) = \frac{2N(j-k+1)[n - N(j-k+1)]}{n-1} \quad (2.3.2)$$

În cazul arborelui binar ( $b=2$ ), densitatea traficului este maximă la nodurile din nivelul al doilea, dar pentru rețelele cu  $b>2$  traficul cel mai dens se manifestă la rădăcină [WITT81]. Dacă distribuția mesajelor se presupune a fi uniformă, majoritatea acestora trec prin căi care conectează două ramuri prin rădăcină (într-un arbore care nu este binar). În această situație, dimensiunea arborelui trebuie să fie limitată pentru a preveni congestiile de trafic de la nodul rădăcină și cele apropiate de acesta. O măsură de reducere a congestiilor constă în prevederea de legături suplimentare între nodurile din același nivel sau dintre nodurile care au adrese care diferă printr-un singur bit, obținându-se astfel rețele de tip hiperarbore (hypertree networks).

Rețelele *complet conectate* (completely connected networks) au câte o legătură între fiecare pereche de noduri, numărul total al lor fiind  $n(n-1)/2$ . Aplicațiile lor se limitează doar la  $n<5$ , deoarece costurile de linii și de conexiuni cresc cu  $n^2$  [WITT81]. Pentru rate fixe de transmisie a mesajelor la toate nodurile, densitatea traficului în rețea, care are valoarea  $2/(n-1)$ , scade vertiginos odată cu creșterea dimensiunii, deoarece sunt mai multe legături decât noduri.

O arie de procesoare bidimensională de tip *plasă* care interconectează vecinii cei mai apropiați (nearest neighbor mesh) se poate obține dacă fiecare nod se conectează prin câte o legătură la cei patru vecini, iar capetele libere pot fi recirculate pe partea opusă. Dacă pe o latură sunt  $w$  procesoare, atunci numărul total de procesoare este  $n=w^2$ , iar în cazul general al unei plase regulate într-un spațiu  $D$ -dimensional acest număr este  $n=w^D$ . Distanța maximă de transmitere a unui mesaj în orice direcție este de  $w/2$ , iar lungimea



medie de transmitere a mesajelor este  $Dw/4$  [WITT81]. Deoarece fiecare nod se conectează în fiecare din cele  $D$  dimensiuni la câte doi vecini, există  $2D$  legături la un nod. Pentru întreaga rețea vor exista în total  $2Dn/2=Dn$  legături. Rezultă că densitatea medie a traficului de mesaje pentru fiecare legătură a rețelei cu  $n$  noduri este:

$$T = \frac{nD(w/4)}{nD} = \frac{w}{4} \quad (2.3.3)$$

Un nod al rețelei trebuie să dispună de  $2D$  porturi de conectare la vecini. Numărul de procesoare poate fi crescut prin mărirea lui  $w$ , dar aceasta duce, așa cum arată relația (2.3.3), la creșterea densității traficului și a întârzierilor de transmitere a mesajelor.

Structura de tip *hipercub* (hypercube) este caracterizată de prezența a  $n=2^D$  procesoare, interconectate sub forma unui cub binar  $D$ -dimensional. Fiecare nod are legături directe și separate la  $D$  alte noduri vecine. Există în total  $2^D$  adrese binare distincte asociate nodurilor, astfel încât adresa unui nod diferă de cea a fiecărui vecin printr-un singur bit. Hiper cubul generalizat (generalized hypercube) are lățimea  $w$  (noduri pe latură), dar păstrează caracteristica de adiacență a adreselor. Numărul total de noduri este  $n=w^D$  ca și la rețeaua de tip plasă, dar aici conectarea legăturilor necesită de două ori mai puține porturi la un nod ( $D$ ) decât la plasă ( $2D$ ). Deoarece fiecare latură se întinde peste  $w$  noduri, în total există  $nD/w$  laturi. Numărul mediu de pași pentru transmiterea unui mesaj este de  $D(w-1)/w$  [WITT81], ca urmare densitatea traficului pe fiecare latură va fi:

$$T = \frac{nD(w-1)/w}{nD/w} = w-1 \quad (2.3.4)$$

Extensia dimensiunii unui hiper cub este mai dificilă decât a unei rețele plasă, deoarece creșterea numărului de noduri care folosesc o latură ( $w$ ) este limitată de traficul admisibil, iar creșterea dimensiunii  $D$  ar presupune mărirea numărului de porturi la fiecare nod. Există și structura în care fiecare vârf al unui hiper cub binar ( $w=2$ ) de dimensiune  $D$  conține o buclă de  $D$  noduri, numită rețea de cicluri conectate ca un cub (cube connected cycles network). În general într-un sistem de tip hiper cub fiecare procesor dispune de memoria sa locală proprie și problemele de congestie sunt mai puțin serioase, dar căile de comunicație serială reprezintă un factor de limitare a debitului. Este posibilă însă și structura în care nodurile conțin memorie comună accesibilă din alte noduri pe legăturile care funcționează ca magistrale, un exemplu fiind rețeaua plasă cu conectivitate suprapusă (overlapping connectivity mesh network) [WILK91].

Rețelele de calculatoare, în care spre deosebire de sistemele multiprocesor calculatoarele dispun de o mare autonomie locală și dedică doar o mică parte din timpul lor pentru activități comune (gestionare de baze de date comune sau acces la anumite periferice scumpe), se bazează de regulă pe rețele statice de interconectare. Astfel pentru *rețele locale de calculatoare* (LAN, local area network) sunt obișnuite topologiile de tip magistrală, inel, stea sau arbore, prezentate mai înainte. Structura de tip magistrală este adecvată pentru tehnici de acces aleator care divizează mediul de transmisie (uzual cablul coaxial) printre sursele din rețea, care transmit mesaje însoțite de adrese recunoscute de stațiile destinație. Rețelele în inel folosesc de obicei metoda de tip "token passing", în care dreptul de a transmite este pasat unui nod în momentul în care recepționează o anumită secvență de cod numită token. După transmiterea mesajului, token-ul este transferat în inel la următorul nod, după o secvență prestabilită. La rețelele în arbore se preferă algoritmi de difuzie, astfel încât un nod părinte transmite mesajul la descendenți, până

când acesta ajunge la ultimul nod. Acesta va răspunde cu o anumită întârziere, iar fiecare părinte colectează răspunsurile de la toți descendenții și numai după aceea va răspunde mai departe. Topologia în stea conectează utilizatorii la capetele legăturilor care pornesc de la un element central (hub). Cel mai frecvent se utilizează ca și mediu de transmisie fibra optică, care este prin esență adecvată unei transmisii de tip punct-la-punct la o rată înaltă de ordinul 100 Mbps [GITL92]. (Includerea de derivații pe fibra optică induce pierderi apreciabile și necesită amplificare optică dacă numărul acestora depășește o anumită limită).

În majoritatea aplicațiilor, rețelele LAN lucrează în medii închise (clădiri) pe o întindere restrânsă, ceea ce face ca fiabilitatea legăturilor să fie de mai mică importanță decât la rețelele care acoperă o arie largă. Fiabilitatea legăturilor trebuie să fie în strânsă corelație cu topologia rețelei. De exemplu, în proiectul rețelei ARPA (Advanced Research Projects Agency), citată frecvent în literatură ca o rețea de referință, și anume o *rețea întinsă de calculatoare* (WAN, wide area network), s-a specificat necesitatea de a exista cel puțin două căi între oricare perechi de noduri. Spre deosebire de aceasta, datorită specificului arătat mai sus, rețelelor LAN le este suficient să permită doar o singură cale între sursă și destinație. Un alt motiv care limitează dimensiunea rețelelor LAN este faptul că lățimea de bandă nu se ridică la valoarea din rețelele largi, mediile de transmisie uzuale fiind destinate pentru viteze de ordinul megabiți pe secundă la un cost relativ scăzut. Datorită cuplajului slab dintre calculatoare, rețelele de calculatoare prezintă o fiabilitate, disponibilitate și un grad de supraviețuire ridicate. Elementele de procesare fiind de regulă identice, specializarea pentru o anumită aplicație fiind asigurată prin software, o unitate defectă poate fi ușor înlocuită cu o alta identică de rezervă. Probabilitatea de a exista un defect în sistem crește însă pe măsură ce numărul de unități active este mai mare, dar o unitate defectă poate fi izolată relativ ușor, permițând funcționarea în continuare a rețelei [PRAD86][KAKAS94d]. Factorii care afectează performanțele unei rețele sunt cantitatea de date și frecvența cu care acestea se transmit, viteza transmisiei și calea pe care o urmează prin rețea. Primii doi factori depind de algoritmul în sine și de partiționarea acestuia, pe când ceilalți doi sunt funcție de hardware.

#### 2.4. Rețele de interconectare dinamice

Într-o *rețea de interconectare dinamică* (dynamic interconnection network) conexiunea dintre două noduri (procesoare sau procesoare/memorii) se stabilește prin intermediul unor comutatoare electronice (active), astfel încât schimbarea stărilor acestora face posibilă înfăptuirea comunicației între diferite perechi de noduri. Rețelele care admit conexiuni simultane între toate combinațiile de intrări și ieșiri se numesc *rețele fără blocare* (non-blocking networks). Acestea sunt strict fără blocare (strictly non-blocking) dacă indiferent de conexiunile existente și fără a le perturba, totdeauna se poate stabili o cale între oricare intrare nefolosită și oricare ieșire nefolosită. Unele rețele fără blocare permit efectuarea unei noi conexiuni intrare/ieșire fără afectarea celor existente numai dacă acestea din urmă au fost înfăptuite pe baza unui anumit algoritm de dirijare prescrist. Astfel de rețele sunt considerate a fi fără blocare într-un sens larg (wide-sense non-blocking). Pentru a reduce numărul de comutatoare, majoritatea rețelelor sunt *rețele cu blocare* (blocking networks), adică nu fac posibile toate combinațiile de conexiuni de intrare/ieșire simultan. Dacă căile blocate pot fi totuși restabilite și redirijate prin modificarea stării comutatoarelor, atunci rețeaua se numește rearanjabilă.

În general comutatoarele sunt grupate pe niveluri, care sunt capabile să interconecteze un număr de intrări cu un număr de ieșiri. În funcție de numărul de niveluri pe care le posedă, rețelele de interconectare dinamice pot fi clasificate în:

- rețele pe un singur nivel
- rețele multinivel

Într-o rețea pe un singur nivel (single stage network), datele transferate trec printr-un singur etaj de comutare de la intrare spre ieșire. Exemplul fundamental de rețea dinamică pe un singur nivel îl reprezintă rețeaua de comutare crossbar (crossbar switch) care poate interconecta  $n$  elemente procesoare (având câte o posibilă memorie locală) cu  $m$  module de memorie comună (figura 2.4.1). Aceasta este o rețea fără blocare și permite accese simultane la toate modulele de memorie, ceea ce înseamnă un debit mare de prelucrare. Un conflict poate apare numai când mai multe procesoare lansează concomitent cereri către aceeași memorie. Întrucât în fiecare cale este implicat numai un singur comutator, întârzierea introdusă de un transfer prin rețea este minimă. Dezavantajul îl constituie creșterea proporțională a costului rețelei cu numărul de comutatoare ( $n \times m$ ), ceea ce reduce domeniul de aplicabilitate la sisteme de dimensiuni mici.

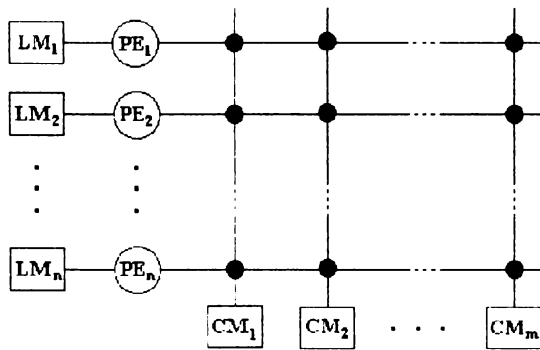


Figura 2.4.1.

Pe un singur nivel se poate realiza o rețea cu amestecare perfectă (perfect shuffle), care intercalează prima jumătate a intrărilor cu cea de a doua jumătate, realizând pentru  $n$  intrări ( $i = 0, 1, \dots, n-1$ ) funcția:

$$\text{sh}(i) = 2i + \lceil 2i/n \rceil \pmod{n} \quad (2.4.1)$$

unde  $\lceil \cdot \rceil$  reprezintă rotunjirea la cel mai mic întreg. Pentru exemplul din figura 2.4.2 ( $n=8$ ), intrările etichetate cu 0, 1, ..., 7 vor da după amestecare secvența 0, 4, 1, 5, 2, 6, 3, 7. Pentru a efectua toate interconexiunile posibile, ieșirile trebuie recirculate la intrări, fiind introduse celule de comutare pentru a se realiza într-un număr de pași conexiunea dorită. Celulele de comutare au câte două intrări și două ieșiri, fiind posibile două stări de transfer (figura 2.4.3). Pentru a se obține toate permutările este necesar un număr de maxim  $n$  recirculări. Aceasta deja corespunde unui transfer al informației prin mai multe etaje de comutatoare, ceea ce caracterizează o rețea de interconectare multinivel (MIN, multistage interconnection network). Fără a mai fi necesară recircularea datelor se poate construi o rețea multinivel prin conectarea în cascadă a  $N = \log_2 n$  rețele cu amestecare

perfectă (figura 2.4.4). Rețeaua obținută este o rețea *Omega*, care are caracteristic un algoritm foarte simplu de autodirijare (self-routing) a informației. La datele utile se atașează o etichetă de dirijare (routing tag), care aici reprezintă tocmai adresa binară a destinației și prin examinarea căreia se va face decizia asupra rutei în fiecare nivel. Cel mai semnificativ bit al adresei se folosește pentru a comanda celula din primul nivel: dacă este 0 atunci se selectează ieșirea superioară, iar dacă este 1 atunci ieșirea inferioară. Următorul bit mai semnificativ al adresei destinație se folosește pentru a selecta ieșirea celulei din următorul nivel, și așa mai departe până când se ajunge la ieșirea finală. Rețeaua Omega este cu blocare, iar proprietatea de autodirijare sugerează o transmisie prin comutare de pachete.

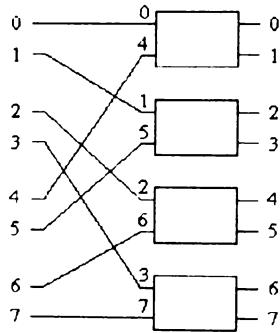


Figura 2.4.2.

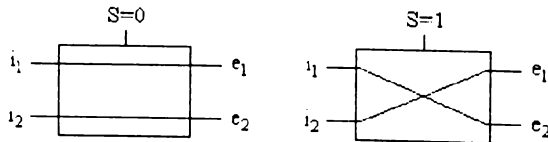


Figura 2.4.3.

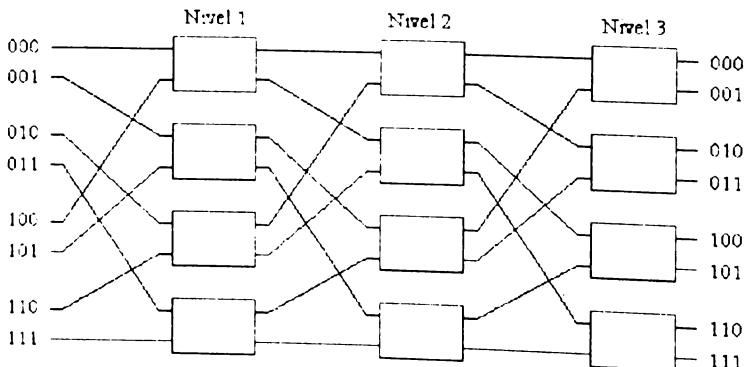


Figura 2.4.4.

Același algoritm de autodirijare descris mai înainte se utilizează și în rețeaua *baseline* arătată în figura 2.4.5. În acest caz fiecare nivel al rețelei înjumătățește succesiv domeniul de dirijare al informației. Și această rețea, ca de altfel majoritatea celor bazate pe celule  $2 \times 2$ , este o rețea cu blocare. Rețelele cu asemenea celule de comutare pot interconecta  $n=2^N$  procesoare, unde  $N$  reprezintă numărul de niveluri. Numărul de celule pe un nivel este  $n/2$ , iar în întreaga rețea  $(n/2) \log_2 n$ , spre deosebire de o rețea crossbar  $n \times n$  în care numărul total de comutatoare este  $n^2$ . Din acest punct de vedere o rețea MIN poate reprezenta o alternativă economică avantajoasă față de o rețea crossbar, în schimb viteza de lucru datorată aici comunicației seriale este mai mică [TABAK90].

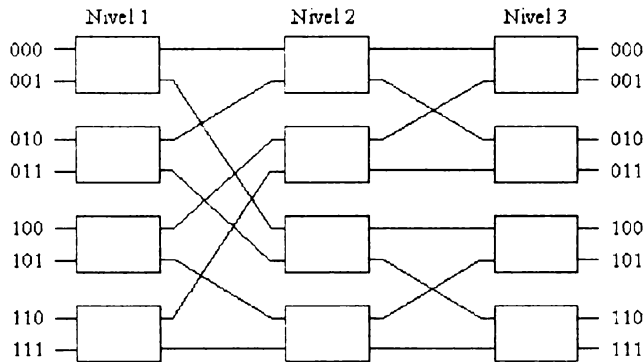


Figura 2.4.5.

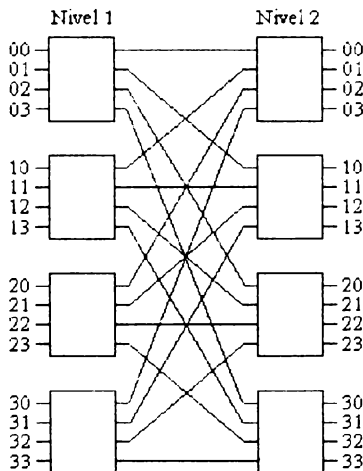


Figura 2.4.6.

Rețelele cu autodirijare pot fi extinse pentru a utiliza o bază de numerație mai mare ca 2 și o amestecare generalizată. De exemplu în figura 2.4.6 este prezentată o rețea Delta de două niveluri cu baza 4, în care adresa destinație este specificată ca un număr în această bază. Fiecare cifră a destinației are o valoare între 0 și 3 care servește la selecția uneia din cele patru ieșiri ale comutatorului crossbar de  $4 \times 4$ . Distribuția legăturilor acestei rețele realizează o amestecare de patru. În cazul mai general, o rețea MIN interconectează  $a^N$  intrări cu  $b^N$  ieșiri prin intermediul unor elemente de comutare crossbar de tip  $a \times b$ , așezate pe  $N$  niveluri. Configurația rețelei pentru  $N=2$  este dată în figura 2.4.7. Dacă se utilizează baze de numerație diferite pentru fiecare nivel, generalizarea se poate extinde, obținându-se rețele cu amestecare generalizate (GSN, generalized shuffle network). Toate aceste rețele MIN permit numai câte o singură rută de la fiecare intrare la fiecare ieșire, nefiind astfel tolerante la defecțiunile celulelor de comutare. Se pot introduce căi alternative între intrări și ieșiri dacă se prevăd niveluri suplimentare în rețea [WILK91].

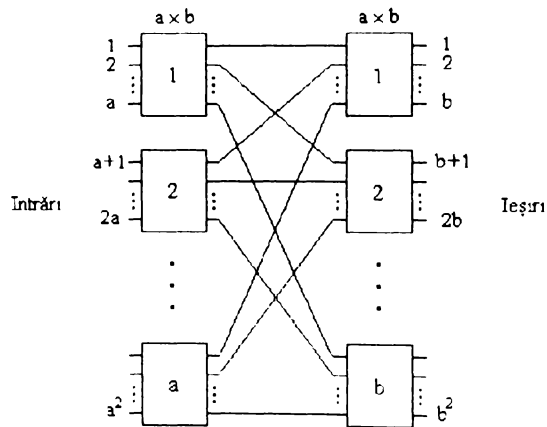


Figura 2.4.7.

Sistemele multiprocesor care conțin un număr mare de elemente de procesare sunt foarte vulnerabile la defecte și se pune problema de a utiliza rețele de interconectare MIN cu proprietăți de toleranță la defecte. O rețea MIN tolerantă la defecte trebuie să furnizeze până la un număr admis de defecte o conexiune sigură între oricare pereche de intrare-ieșire. Există mai multe căi de a face un MIN tolerant la defecte. Unele metode utilizează coduri corectoare de eroare [VLAD86][VLAD89] pe căile de transmisie, altele introduc conexiuni redundante pentru fiecare pereche de intrare-ieșire. O altă cale de creștere a fiabilității sistemului este adăugarea de procesoare PE de rezervă, fiind astfel posibilă și luarea în considerare a defecțiunilor la nivelul PE, urmată de reconfigurarea sistemului. O metodă de construcție a unor astfel de rețele cu redundanță dinamică DR (dynamic redundancy networks) este prezentată în lucrarea [JENG88].

Metoda pornește de la o rețea MIN cubică cu  $N = 2^m$  intrări/ieșiri conținând  $N/2$  comutatoare  $2 \times 2$  dispuse pe  $m$  niveluri. La setul inițial de  $N$  procesoare PE se adaugă  $S$  de rezervă, în situația  $S = 2^r$  fiind posibilă obținerea unei rețele DR reduse (RDR) cu un număr mai mic de interconexiuni. Un exemplu pentru  $N = 8$  și  $S = 4$  este prezentat în figura 2.4.8.

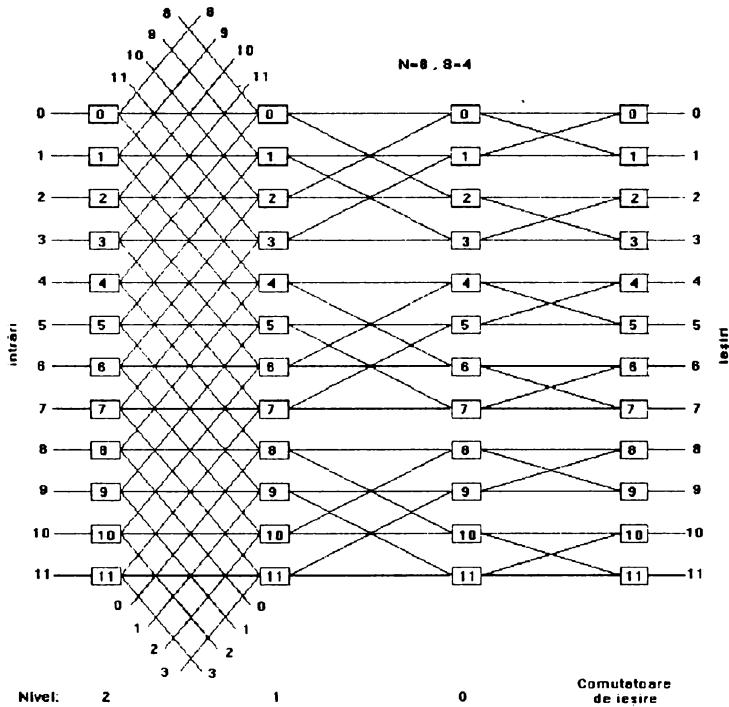


Figura 2.4.8.

Toleranța la defecte a rețelei DR este asigurată prin reconfigurare. În caz de detectare a unui defect va porni un proces de reconfigurare, care duce la eliminarea din partea activă a rețelei a rândului pe care a apărut defectul și a procesorului de pe rândul respectiv. Un rând conține toate comutatoarele de rețea cu aceeași adresă, toate legăturile incidente de intrare și ieșire la acestea. Pentru exemplul dat, dacă se defectează  $PE_1$ , se vor izola procesoarele și rândurile 0, 1, 2 și 3, cele patru procesoare de rezervă devenind active. Se observă că performanțele sistemului nu se degradează după reconfigurare, dar este în sarcina sistemului de operare de a redistribui funcțiile și de a reporni sistemul dintr-un punct anterior în care taskurile au fost lipsite de eroare (backward recovery).

Pentru transferul datelor între procesoare, acestea adaugă și în acest caz la informația utilă etichete de dirijare, care determină poziționarea corespunzătoare a comutatoarelor de pe niveluri.

O altă soluție de rețea MIN tolerantă la defecte este prezentată în [TZENG88]. Aceasta se bazează pe găsirea de căi alternative în fiecare nivel în cazul întâlnirii unui defect, o rețea cu astfel de proprietate fiind numită *rețea cu redirijare puternică* (strong reroutable network). Realizarea practică constă în conectarea în lanț a tuturor elementelor de comutare SE (switching element) aflate pe același nivel al unui arbore. Câte un arbore se poate forma de la fiecare ieșire a rețelei de bază (considerată ca rădăcină) către toate intrările, așa cum se arată cu linii groase în figura 2.4.9. Pentru conectarea în lanț,

elementele de comutare vor fi prevăzute suplimentar cu câte o intrare și o ieșire, ele transformându-se astfel din comutatoare crossbar de tip  $2 \times 2$  în comutatoare de tip  $3 \times 3$ .

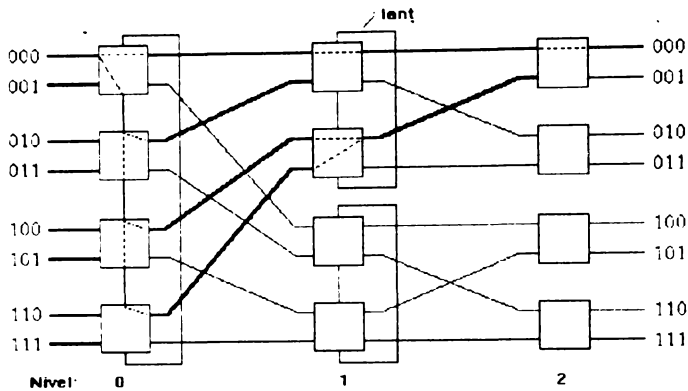


Figura 2.4.9.

Algoritmul de dirijare al datelor în rețeaua cu înlănțuire rămâne în esență cea introdusă de Lawrie [LAWR75] bazată pe etichetele de dirijare, care a fost prezentată mai înainte. Eticheta destinație este reprezentată de însăși valoarea binară a ieșirii  $d_0d_1d_2$ , astfel încât în nivelul  $i$  dacă  $d_i = 0$  atunci SE comută pe ieșirea superioară, iar dacă  $d_i = 1$  atunci pe ieșirea inferioară. Dacă ieșirea selectată este blocată ca urmare a unui defect pe ieșire, pe linia de legătură sau pe intrarea elementului SE de pe următorul nivel la care este conectată ieșirea în cauză, atunci cererea este dirijată prin ieșirea de conectare în lanț către un alt SE din lanț. Cum se observă pe traseele cu linie întreruptă din figură, același bit  $d_i$  va fi utilizat de un algoritm de dirijare identic la noul SE, ceea ce reprezintă un avantaj. Pe ultimul nivel nu are sens înlănțuirea unui element SE cu el însuși, deoarece acest lucru nu mai duce la creșterea performanțelor, ci numai la blocarea cererii.

Făcând o analiză comparativă a rețelelor de interconectare prezentate, se poate spune că în ciuda congestiei de date și a fiabilității scăzute, sistemele orientate pe magistrale sunt răspândite datorită costului redus și a configurabilității ridicate. Rețelele crossbar, cu toate că prezintă o viteză mare de lucru, sunt mai rar utilizate din cauza complexității și costului ridicat, precum și datorită dificultăților de reconfigurare. În schimb rețelele multinivel, moderate ca și cost, produc simultan un număr de căi de comunicație, ceea ce reduce timpul total de așteptare pentru efectuarea unui transfer de date față de un sistem cu magistrală unică. Pe lângă creșterea capacității de prelucrare a fiecărui procesor, reducerea costurilor de comunicare fără afectarea sensibilă a prelucrării paralele, reprezintă un deziderat în proiectarea sistemelor multiprocesor, ceea ce depinde în mare măsură de rețeaua de comunicație aleasă.



### 3. FIABILITATEA SISTEMELOR CU COMPONENTE MULTIPLE

#### 3.1. Noțiunea de fiabilitate

Într-un sens mai larg, fiabilitatea poate fi considerată o parte componentă a calității unui produs. Calitatea unui produs se exprimă printr-un set de descriptori care conțin mărimi referitoare la caracteristicile fizice ale acestuia și atribute care arată în ce măsură produsul corespunde cerințelor utilizatorului. Aceste atribute funcționale duc la luarea unei decizii de tip “go/no go” asupra produsului, în sensul că el este considerat acceptabil sau deficient. În context calitativ, fiabilitatea reprezintă o măsură a cum produsul îndeplinește specificațiile impuse de funcționare pe o anumită perioadă de timp. Caracteristica studiată de teoria fiabilității o reprezintă durata de viață a produsului aflat în exploatare, exprimând conservarea în timp a performanțelor produsului sub forma unei calități dinamice.

În sens mai restrâns, fiabilitatea este o mărime cantitativă care exprimă probabilitatea ca un produs să fie operațional. Prin produs se poate înțelege o componentă singulară, un sistem compus din mai multe componente sau chiar un program de calculator. Analiza de fiabilitate ține cont de apariția unor evenimente neanticipate și nedorite în cursul exploatării produsului, care au influență asupra comportării sale. Aceste evenimente pot fi defectări ale componentelor sau modificări ale parametrilor acestora în afara limitelor admisibile cauzate de condițiile de mediu sau uzură, precum și problemele de proiectare și tehnologice care ies la iveală în timpul utilizării. Problema de seamă o constituie faptul că două produse identice care lucrează în condiții similare nu se defectează de regulă în același moment. Acest fenomen poate fi descris numai în termeni probabilistici. Durata de viață utilă (sau timpul până la defectare) constituie o *variabilă aleatoare*  $T$ , a cărei lege probabilistică se poate exprima în mai multe feluri. Este de subliniat faptul că în această accepțiune produsul este caracterizat printr-o comportare bivalentă, cu două stări: funcționează sau este defect. Starea defectă semnifică faptul că el nu este în stare să satisfacă funcțiile cerute.

**Definiția 3.1.1.** *Fiabilitatea*  $R(t)$  (reliability) a unui produs (componentă sau sistem) se definește ca fiind probabilitatea ca acesta lucrând în condiții date de mediu, să-și îndeplinească adecvat funcțiile specificate pe durata unui interval dat de timp  $[0, t]$ :

$$R(t) = \Pr(T > t) \quad (3.1.1)$$

Se observă că definiția fiabilității presupune următoarele aspecte:

- Admiterea posibilității de defectare, adică a unei noțiuni probabilistice
- Acceptarea unui nivel de performanță precizat de proiectant sau de utilizator
- Exploatarea produsului în condiții de funcționare prescrise

Funcția  $R(t)$  se mai numește *funcție de supraviețuire* în cazul sistemelor monotone, caracterizate prin ireversibilitatea defectărilor [TĂRC89].

Legea de defectare probabilistică a unui produs se exprimă adesea prin *funcția de distribuție cumulată*  $F(t)$ , definită prin probabilitatea ca produsul să funcționeze cel mult până în momentul  $t$ :

$$F(t) = \Pr(T \leq t) \quad (3.1.2)$$

Aceasta mai este referită ca *funcția de defectare* sau de *nefiabilitate* (unreliability), fiind complementară funcției de fiabilitate:

$$F(t) = \Pr(T \leq t) = 1 - \Pr(T > t) = 1 - R(t) \quad (3.1.3)$$

Pe baza acestei definiții și având în vedere faptul că la momentul inițial produsul este cu certitudine funcțional,  $F(t)$  este o funcție monotonă nedescrescătoare astfel încât  $F(0)=0$  și  $\lim_{t \rightarrow \infty} F(t) = 1$ . Ca urmare a relației (3.1.3) rezultă imediat că  $R(0)=1$  și  $\lim_{t \rightarrow \infty} R(t) = 0$ .

Caracterizări statistice ale duratei de viață sunt date de  *timpul mediu până la defectare*  $MTTF$  (mean time to failure) și de *dispersia relativă a timpului de defectare*  $RFTD$  (relative failure time dispersion). Fiind cunoscută legea de distribuție pentru  $T$ ,  $MTTF$  se calculează ca valoarea așteptată (expected value), adică media (mean) variabilei aleatoare,

$$MTTF = \mu = E[T] \quad (3.1.4)$$

iar  $RFTD$  reprezintă deviația standard  $\sigma$  a lui  $T$  raportată la media sa (adică coeficientul de variație al variabilei aleatoare):

$$RFTD = \frac{\sigma}{\mu} = \frac{\sqrt{E[(T - \mu)^2]}}{E[T]} \quad (3.1.5)$$

Prin deviație standard (standard deviation) sau abaterea medie pătratică a variabilei aleatoare se înțelege rădăcina pătrată a dispersiei sau varianței (variance), care caracterizează gradul de împrăștiere a valorilor variabilei aleatoare în jurul valorii medii:

$$\text{Var}[T] = \sigma^2 = E[(T - \mu)^2] \quad (3.1.6)$$

Este de remarcat faptul că atât  $MTTF$  cât și  $RFTD$  reprezintă caracteristici ale repartiției timpului de defectare  $T$ , fiind independente de timpul de observare  $t$ . În cazul în care se cere să se compare două sisteme din punct de vedere al fiabilității lor, este convenabil să se utilizeze un indicator comparativ cum ar fi *factorul de îmbunătățire al fiabilității*  $RIF$  (reliability improvement factor) sau *factorul de îmbunătățire al timpului de misiune* (mission time improvement factor) [VISW87]. Timpul de misiune al unui produs nu trebuie confundat cu durata de viață a acestuia, ci este definit în felul următor:

**Definiția 3.1.2.** *Timpul de misiune* (mission time) al unui produs reprezintă durata maximă  $T_{\text{MAX}}$  pentru care fiabilitatea produsului respectă o limită specificată  $R_{\text{MIN}}$ , adică pentru  $t \leq T_{\text{MAX}}$  funcția de fiabilitate  $R(t) > R_{\text{MIN}}$ .

Pentru a determina legea de repartiție a timpului de defectare se efectuează încercări de fiabilitate asupra unui lot de produse identice. Măsurătorile produc estimatori de fiabilitate, care se prelucrează cu metode statistice [GROSH89][GEBER84] [HOHAN82].

Întrucât prezenta teză abordează probleme de fiabilitate ale sistemelor de calcul, voi trata în continuare pe lângă cazul distribuției de tip continuu a timpului de defectare și cazul distribuției discrete, aducând cu această ocazie o abordare personală, problematica nefiind dezvoltată în cărțile clasice de fiabilitate. Acest lucru este justificat de faptul că procesoarele au o funcționare compusă din cicluri instrucțiune, care pot fi considerate ca segmente discrete de timp.

### 3.2. Fiabilitatea în spațiul timpului continuu

În acest caz variabila aleatoare  $T$  corespunzătoare momentului de apariție a defectului poate lua orice valoare pe axa timpului  $t \geq 0$ .

Admițând că  $F(t)$  constituie conform (3.1.2) probabilitatea ca variabila aleatoare să ia valoare mai mică sau egală cu  $t$ , legea de repartiție a variabilei aleatoare se exprimă adesea prin viteza instantanee de defectare  $f(t)$ . Aceasta reprezintă probabilitatea de defectare pe unitatea de timp, adică:

$$\begin{aligned} f(t) &= \lim_{\Delta t \rightarrow 0} \frac{\Pr(t \leq T \leq t + \Delta t)}{\Delta t} = \lim_{\Delta t \rightarrow 0} \frac{\Pr(T \leq t + \Delta t) - \Pr(T \leq t)}{\Delta t} = \\ &= \lim_{\Delta t \rightarrow 0} \frac{F(t + \Delta t) - F(t)}{\Delta t} = \frac{dF(t)}{dt} \end{aligned} \quad (3.2.1)$$

În termeni probabilistici, funcția  $f(t) \geq 0$  se numește *funcția de densitate de probabilitate* (probability density function) și are proprietatea:

$$\int_0^{\infty} f(t) dt = 1 \quad (3.2.2)$$

Este de menționat faptul că  $f(t)$  nu este o probabilitate, însă cantitatea  $f(t)dt$  este interpretată ca elementul de probabilitate. Pe baza celor prezentate, rezultă că *funcția de distribuție cumulată* (CDF, cumulative distribution function), o funcție continuă, se exprimă ca

$$F(t) = \int_0^t f(u) du \quad (3.2.3)$$

și pe grafic reprezintă aria de sub curba de densitate până la abscisa considerată (figura 3.2.1). Pe baza complementarității exprimată de relația (3.1.3), este evident că aria din dreapta corespunde fiabilității  $R(t)$ :

$$R(t) = \int_t^{\infty} f(u) du \quad (3.2.4)$$

De asemenea, se verifică imediat și egalitățile:

$$f(t) = F'(t) = -R'(t) \quad (3.2.5)$$

Fiind dat  $f(t)$ , probabilitatea ca variabila aleatoare să aparțină intervalului  $[t_1, t_2]$  se poate calcula prin integrare:

$$\Pr(t_1 \leq T \leq t_2) = F(t_2) - F(t_1) = \int_{t_1}^{t_2} f(t) dt \quad (3.2.6)$$

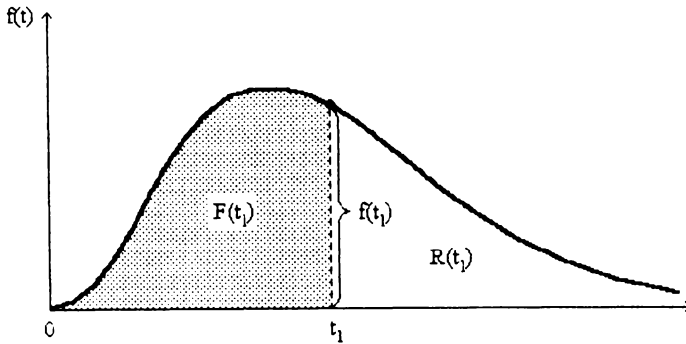


Figura 3.2.1.

Probabilitatea ca variabila aleatoare să ia o anumită valoare, deci ca defectarea să aibă loc la un anumit moment de timp este însă zero, deoarece:

$$\Pr(T = t_1) = \Pr(t_1 \leq T \leq t_1) = \int_{t_1}^{t_1} f(t) dt = 0 \quad (3.2.7)$$

De aici rezultă și egalitățile:

$$\Pr(t_1 \leq T \leq t_2) = \Pr(t_1 \leq T < t_2) = \Pr(t_1 < T \leq t_2) = \Pr(t_1 < T < t_2) \quad (3.2.8)$$

Există încă o posibilitate de caracterizare a legii la care se supune timpul până la defectare, utilizând probabilități condiționate. Din teoria probabilităților este cunoscută formula de calcul a probabilității producerii unui eveniment  $A$ , cu condiția ca să se producă și evenimentul  $B$ :

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)}$$

Aici  $A$  semnifică evenimentul "sistemul se defectează în intervalul  $[t, t + \Delta t]$ ", iar  $B$  evenimentul "sistemul funcționează până în momentul  $t$ ". Ca urmare,  $\Pr(A \cap B) = F(t + \Delta t) - F(t)$ , iar  $\Pr(B) = 1 - F(t)$ . Prin definiție, funcția de hazard (hazard function)  $h(t)$  reprezintă

probabilitatea de defectare pe unitatea de timp la momentul  $t$ , cu condiția ca să nu fi apărut defecțiunea până la momentul  $t$ , deci:

$$h(t) = \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \frac{F(t + \Delta t) - F(t)}{1 - F(t)} = \frac{F'(t)}{1 - F(t)} = -\frac{R'(t)}{R(t)} = \frac{f(t)}{R(t)} \quad (3.2.9)$$

Funcția de hazard reprezintă o densitate de probabilitate condiționată și în context fiabilistic se mai numește *rata de defectare* (failure rate) sau *intensitatea de defectare*. Cantitatea  $h(t)$  nu are semnificație grafică clară pe figura 3.2.1, fiind ordonată împărțită la aria din dreapta sa.

Se poate constata că fiind dat oricare din indicatorii de fiabilitate  $f(t)$ ,  $F(t)$ ,  $R(t)$  sau  $h(t)$  pentru a caracteriza repartiția timpului de defectare, ceilalți trei se pot deduce din acesta. O sistematizare a formulilor de calcul este dată în [CĂȚU89a]. Cazul când este dat  $h(t)$  necesită un calcul mai greu,  $R(t)$  fiind obținut ca soluția unei ecuații diferențiale [GROSH89]:

$$R(t) = e^{-\int_0^t h(u) du} = e^{-H(t)} \quad (3.2.10)$$

Relația (3.2.10) este cunoscută și ca formula exponențială a fiabilității.

Funcția  $H(t)$  se numește *funcția de hazard cumulată* (cumulative hazard function) și se poate scrie ca și o funcție de supraviețuire logaritmică :

$$H(t) = -\ln R(t) \quad (3.2.11)$$

În afară de cei patru indicatori de fiabilitate menționați mai sus, în practică este convenabil să se lucreze cu valoarea medie și dispersia variabilei aleatoare  $T$ . Valoarea medie, adică  $MTTF$ , reprezintă conform (3.1.4) valoarea așteptată a lui  $T$  și se calculează cu formula mediei:

$$\mu = E[T] = \int_0^{\infty} t f(t) dt \quad (3.2.12)$$

**Teorema 3.2.1.** Timpul până la defectare fiind o variabilă aleatoare continuă, media ei ( $MTTF$ ) este dată de aria de sub curba de fiabilitate, adică:

$$MTTF = \mu = \int_0^{\infty} R(t) dt \quad (3.2.13)$$

*Demonstrație.* Înlocuind relația (3.2.5) în (3.2.12), se obține:

$$\int_0^{\infty} t f(t) dt = -\int_0^{\infty} t dR(t)$$

Voi recurge la integrarea prin părți, notând  $u=t$ ,  $dv=dR(t)$ ,  $du=dt$ ,  $v=R(t)$ . Rezultă:

$$\int_0^{\infty} t dR(t) = tR(t)|_0^{\infty} - \int_0^{\infty} R(t) dt = \lim_{t \rightarrow \infty} tR(t) - 0R(0) - \int_0^{\infty} R(t) dt$$

Știind că  $R(0)=1$ , iar  $\lim_{t \rightarrow \infty} \frac{R(t)}{t} = 0$  deoarece se poate admite că  $R(t)$  tinde mai repede către 0 decât  $1/t$  atunci când  $t \rightarrow \infty$  pentru distribuțiile uzuale [GROSH89], se obține rezultatul căutat:

$$\int_0^{\infty} t f(t) dt = \int_0^{\infty} R(t) dt$$

□

Dispersia (varianța) variabilei aleatoare este dată conform (3.1.6) de valoarea așteptată a pătratului abaterii:

$$\begin{aligned} \sigma^2 = E[(T - \mu)^2] &= \int_0^{\infty} (t - \mu)^2 f(t) dt = \int_0^{\infty} t^2 f(t) dt - \\ &- 2\mu \int_0^{\infty} t f(t) dt + \mu^2 \int_0^{\infty} f(t) dt = E[T^2] - \mu^2 \end{aligned} \quad (3.2.14)$$

În evaluarea expresiei precedente am ținut cont de (3.2.2) și (3.2.12).

Dintre legile de distribuție continue cu aplicație în teoria fiabilității amintesc aici distribuția exponențială negativă (cu rata constantă a defectărilor), distribuția normală (utilă în prelucrări statistice, fiind caracteristică mediei unui eșantion de  $n$  variabile independente), distribuția Weibull (cea mai des utilizată distribuție parametrică), distribuția Rayleigh (cu rata linear crescătoare a defectărilor) etc. [JAIN91][TĂRC89][VISW87].

### 3.3. Fiabilitatea în spațiul timpului discret

În acest caz variabila aleatoare  $T$  (timpul până la defectare) poate lua numai valori discrete pe axa timpului, adică  $t = t_0, t_1, t_2, \dots, t_i, \dots$ , în particular  $t = 0, 1, 2, 3, \dots, i, \dots$ .

Mulțimea valorilor variabilei aleatoare fiind discretă, în locul densității de probabilitate de la cazul precedent, distribuția este caracterizată prin *funcția de masă de probabilitate* (probability mass function),  $f(t)$ . Aceasta este însăși o probabilitate, și anume probabilitatea ca variabila aleatoare să ia valoarea  $t$ , adică defectarea să se producă la timpul (discret)  $t$ :

$$f(t) = \Pr(T=t) \quad (3.3.1)$$

Distribuția se poate da sub forma unui tabel de repartiție a variabilei aleatoare, de forma

$$\begin{pmatrix} 0 & t_1 & t_2 & & t_i & \dots \\ 0 & p_1 & p_2 & & p_i & \dots \end{pmatrix}$$

unde  $f(t)=p_i$  reprezintă probabilitatea de defectare în momentul  $t_i$ . În acest tablou am ținut cont de faptul că în momentul inițial sistemul este sigur funcțional, deci  $f(0)=0$ . Din punct de vedere grafic, funcția  $f(t)$  se reprezintă prin puncte numai acolo unde ea este definită (figura 3.3.1). Similar relației (3.2.2), aici se poate scrie:

$$\sum_{i=1}^{\infty} p_i = 1 \quad (3.3.2)$$

Ca și la cazul continuu, și aici se poate defini *funcția de distribuție cumulată* (CDF, cumulative distribution function), care însă nu este continuă și deci nici derivabilă. Prin definiție:

$$F(t) = \sum_{t_i \leq t} p_i \quad (3.3.3)$$

Caracterul discontinuu, în scară, al funcției  $F(t)$ , precum și faptul că  $\lim_{t \rightarrow \infty} F(t) = 1$  se observă pe graficul din figura 3.3.2. Semnificația funcției de distribuție este probabilitatea ca variabila aleatoare să ia valoare mai mică ca  $t$ , adică se poate interpreta ca fiind funcția de defectare conform relației (3.1.2). Relația de complementaritate (3.1.3) cu funcția de fiabilitate este evident valabilă. Probabilitatea ca defectarea să se producă în intervalul  $(t_m, t_n]$  se calculează prin însumare:

$$\Pr(t_m < T \leq t_n) = F(t_n) - F(t_m) = \sum_{t_m < t_i \leq t_n} p_i \quad (3.3.4)$$

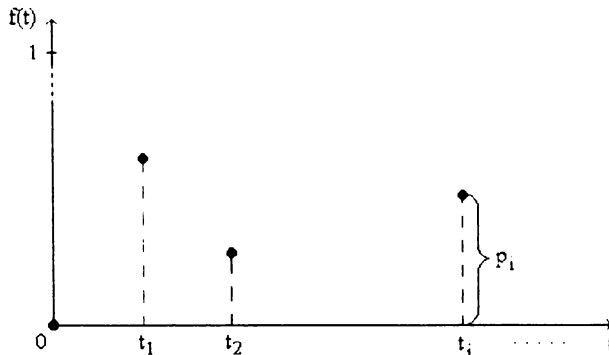


Figura 3.3.1.

Dacă este dată funcția de distribuție cumulată, atunci pentru orice punct  $t$  se poate calcula probabilitatea  $f(t)$  ca variabila aleatoare să ia tocmai această valoare. Pornind de la relația de definiție (3.3.3) se poate scrie

$$F(t_i) = F(t_{i-1}) + f(t_i)$$

de unde:

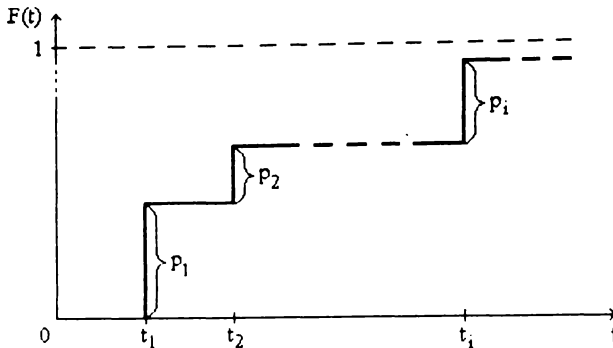


Figura 3.3.2.

$$f(t_i) = F(t_i) - F(t_{i-1}) \quad (3.3.5)$$

Țimpul mediu până la defectare (*MTTF*) este dat de valoarea medie (așteptată) a variabilei aleatoare  $T$ , care în acest caz se calculează ca o sumă ponderată:

$$\mu = E[T] = \sum_{i=1}^{\infty} p_i t_i \quad (3.3.6)$$

În situația în care pentru caracterizarea variabilei aleatoare este dată funcția de distribuție cumulată  $F(t)$  și nu  $f(t)$ , atunci pentru a calcula valoarea așteptată este utilă următoarea leamnă.

**Lema 3.3.1.** Fiind dată funcția de distribuție cumulată  $F(t)$  a variabilei aleatoare discrete  $T$ , care ia valori în mulțimea  $t = t_1, t_2, \dots, t_n$ , valoarea așteptată (media) variabilei aleatoare se calculează cu formula:

$$E[T] = t_n F(t_n) - \sum_{i=1}^{n-1} (t_{i+1} - t_i) F(t_i) \quad (3.3.7)$$

*Demonstrație.* Având în vedere că  $f(t_i) = p_i$  și înlocuind (3.3.5) în (3.3.6), se obține:

$$E[T] = \sum_{i=1}^n t_i [F(t_i) - F(t_{i-1})].$$

cu mențiunea că  $F(t_0) = 0$ , deoarece  $F(t_1) = f(t_1)$ . Dezvoltând relația de mai sus și grupând termenii asemenea, rezultă formula propusă pentru demonstrație:

$$\begin{aligned} E[T] &= t_1 F(t_1) + t_2 F(t_2) - t_2 F(t_1) - t_3 F(t_3) + t_3 F(t_2) + \dots + t_n F(t_n) - t_{n-1} F(t_{n-1}) = \\ &= t_n F(t_n) - (t_2 - t_1) F(t_1) - (t_3 - t_2) F(t_2) - \dots - (t_n - t_{n-1}) F(t_{n-1}) = t_n F(t_n) - \sum_{i=1}^{n-1} (t_{i+1} - t_i) F(t_i) \end{aligned}$$

□



**Caz particular.** Dacă variabila aleatoare discretă este timpul, și anume  $t=1,2,3, \dots$ , atunci relația (3.3.7) devine:

$$E[T] = \lim_{n \rightarrow \infty} \left[ F(n) - \sum_{i=1}^{n-1} F(i) \right] \quad (3.3.8)$$

În continuare voi statua o teoremă, care va fi utilă pentru calculul timpului mediu de bună funcționare, atunci când se cunoaște funcția de fiabilitate a sistemului dată sub formă polinomială.

**Teorema 3.3.2.** Dacă funcția de fiabilitate a unui sistem alcătuit din componente identice, este un polinom de forma

$$R = a_0 + a_1 r + a_2 r^2 + \dots + a_{k-1} r^{k-1}$$

unde  $a_i \in \mathbb{Z}$ ,  $k \in \mathbb{N}$  și  $r=p^t$  reprezintă fiabilitatea unei componente conform unei distribuții geometrice ( $t=1,2,3, \dots$ ),  $q=1-p$  fiind probabilitatea de defectare în unitatea de timp, atunci valoarea așteptată a variabilei aleatoare este:

$$E[T] = 1 - a_0 + a_1 \frac{p}{1-p} + a_2 \frac{p^2}{1-p^2} + \dots + a_{k-1} \frac{p^{k-1}}{1-p^{k-1}} \quad (3.3.9)$$

*Demonstrație.* Funcția de distribuție cumulată este:

$$F(t) = 1 - R(t) = 1 - a_0 - a_1 p^t - a_2 p^{2t} - \dots - a_{k-1} p^{(k-1)t}$$

Pe baza relației (3.3.8), valoarea medie se poate scrie ca:

$$\begin{aligned} E[T] &= \lim_{n \rightarrow \infty} \left[ n(1 - a_0 - a_1 p^n - a_2 p^{2n} - \dots - a_{k-1} p^{(k-1)n}) - \sum_{i=1}^{n-1} (1 - a_0) + \right. \\ &\quad \left. + \sum_{i=1}^{n-1} (a_1 p^i + a_2 p^{2i} + \dots + a_{k-1} p^{(k-1)i}) \right] = \\ &= \left[ n - na_0 - a_1 np^n - a_2 np^{2n} - \dots - a_{k-1} np^{(k-1)n} - (n-1)(1 - a_0) + \right. \\ &\quad \left. + a_1 \sum_{i=1}^{n-1} p^i + a_2 \sum_{i=1}^{n-1} p^{2i} + \dots + a_{k-1} \sum_{i=1}^{n-1} p^{(k-1)i} \right] = \\ &= 1 - a_0 - \lim_{n \rightarrow \infty} (a_1 np^n + a_2 np^{2n} + \dots + a_{k-1} np^{(k-1)n}) + \\ &\quad + \lim_{n \rightarrow \infty} \left( a_1 \sum_{i=1}^{n-1} p^i + a_2 \sum_{i=1}^{n-1} p^{2i} + \dots + a_{k-1} \sum_{i=1}^{n-1} p^{(k-1)i} \right) \end{aligned} \quad (3.3.10)$$

Pentru a evalua prima paranteză este necesară calcularea unor limite de forma  $\lim_{n \rightarrow \infty} (np^{(k-1)n})$ , care este o nedeterminare de tipul  $\infty \cdot 0$ ,  $p$  fiind subunitar. Este necesară aplicarea regulii lui l'Hospital, care duce la:

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{n}{p^{-(k-1)n}} &= \lim_{n \rightarrow \infty} \frac{(n)'}{(p^{-(k-1)n})'} = \lim_{n \rightarrow \infty} \frac{1}{-(k-1)p^{-(k-1)n} \ln p} = \\ &= -\frac{1}{(k-1) \ln p} \lim_{n \rightarrow \infty} p^{(k-1)n} = 0. \end{aligned} \quad (3.3.11)$$

Termenul de forma  $\sum_{i=1}^{n-1} p^{(k-1)i}$  este o progresie geometrică cu rata  $p^{k-1}$ , a cărei sumă este cunoscută:

$$\lim_{n \rightarrow \infty} \sum_{i=1}^{n-1} p^{(k-1)i} = \lim_{n \rightarrow \infty} \left[ \frac{1 - p^{(k-1)n}}{1 - p^{k-1}} - 1 \right] = \frac{1}{1 - p^{k-1}} - 1 \quad (3.3.12)$$

Înlocuind (3.3.11) și (3.3.12) în (3.3.10), se obține relația căutăată:

$$\begin{aligned} E[T] &= 1 - a_0 + a_1 \left( \frac{1}{1-p} - 1 \right) + a_2 \left( \frac{1}{1-p^2} - 1 \right) + \dots + a_{k-1} \left( \frac{1}{1-p^{k-1}} - 1 \right) = \\ &= 1 - a_0 + a_1 \frac{p}{1-p} + a_2 \frac{p^2}{1-p^2} + \dots + a_{k-1} \frac{p^{k-1}}{1-p^{k-1}} \end{aligned}$$

□

**Observație.** Dacă  $a_0=0$ , atunci  $\sum_{i=1}^{n-1} a_i = 1$  deoarece sistemul este considerat a fi bun la început,  $R(0)=1$ .

Dispersia (variația) variabilei aleatoare se calculează similar cu (3.2.14), însă în loc de integrare se folosește însumarea. Ținând cont și de (3.3.6), respectiv (3.3.2), se obține:

$$\begin{aligned} \sigma^2 &= E[(T - \mu)^2] = \sum_{i=1}^{\infty} p_i (t_i - \mu)^2 = \sum_{i=1}^{\infty} p_i t_i^2 - 2\mu \sum_{i=1}^{\infty} p_i t_i + \mu^2 \sum_{i=1}^{\infty} p_i = \\ &= E[T^2] - \mu^2 \end{aligned} \quad (3.3.13)$$

În acest paragraf am prezentat modelarea timpului până la defectare cu ajutorul unei legi de distribuție cu variabilă aleatoare discretă. Utilizarea distribuției de tip discret în studiul performanțelor și fiabilității unui sistem de calcul acoperă însă o gamă foarte largă, din care voi da în continuare câteva exemple.

Distribuția Bernoulli, cea mai simplă distribuție discretă, se folosește pentru a exprima dacă [JAIN91]:

- un sistem este funcțional sau defect
- un pachet de date ajunge sau nu la destinație într-o rețea de calculatoare
- un bit din pachetul de date ajunge afectat de eroare

Distribuția binomială poate modela:

- numărul de procesoare care sunt operaționale într-un sistem multiprocesor
- numărul de pachete care ajung la destinație fără pierderi
- numărul de biți neafecți de eroare dintr-un pachet

Distribuția geometrică are aplicații în determinarea:

- numărului de accese locale între două accese succesive la distanță
- numărului de pachete transmise cu succes între cele care au necesitat retransmitere
- numărului de biți nealterați aflați între biții afectați de eroare la recepția unui pachet

Distribuția Poisson, numită și legea evenimentelor rare, se folosește la modelarea:

- numărului de componente defecte în unitatea de timp
- numărului de accese la un server într-un interval de timp  $t$
- numărului de solicitări la o bază de date în  $t$  secunde

Distribuția uniformă discretă se utilizează în simulări, pentru a desemna:

- numărul pistei pentru o căutare pe disc
- numărul dispozitivului periferic pentru următoarea operație de intrare/ieșire
- nodul sursă și destinație pentru următorul pachet transmis în rețea

### 3.4. Distribuții fără memorarea trecutului

Comportarea ratei de defectare (a funcției de hazard) a unei componente în funcție de timp, numită și caracteristică de viață sau curbă de mortalitate, are aspectul de “cadă de baie” (bathtub curve) din figura 3.4.1. Curba prezintă trei zone distincte. Prima perioadă (A), de rodaj (burn-in) sau mortalitate infantilă (infant mortality), pune în evidență defecțiunile precoce, cu o rată de defectare în scădere. Porțiunea B reprezintă viață utilă (useful life), cu rata de defectare considerată constantă, iar etapa finală (C) prezintă creșterea ratei de defectare corespunzătoare uzurii (wear out) și îmbătrânirii. Este de menționat faptul că pentru produse reparabile, curba nu mai prezintă zona cu rata de defectare constantă [VISW87].

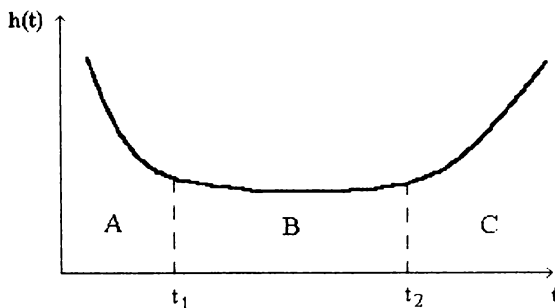


Figura 3.4.1.

Curba se poate descrie matematic cu o funcție polinomială de forma  $h(t) = c_1 t^{c_2}$ , unde  $c_1 > 0$ , iar pentru  $c_2 < 0$  se obține funcția de hazard descrescătoare, pentru  $c_2 = 0$  constantă și pentru  $c_2 > 0$  crescătoare. Notând  $c_1 = \lambda\beta$  și  $c_2 = \beta - 1$  se obține forma cunoscută a funcției de hazard pentru distribuția Weibull:

$$h(t) = \lambda \beta t^{\beta-1} \quad (3.4.1)$$

Funcția de repartiție și densitatea de probabilitate sunt date, ca urmare a relațiilor (3.2.10) și (3.2.5) de:

$$F(t) = 1 - e^{-\int_0^t \lambda \beta u^{\beta-1} du} = 1 - e^{-\lambda t^\beta} \quad (3.4.2)$$

$$f(t) = \lambda \beta t^{\beta-1} e^{-\lambda t^\beta} \quad (3.4.3)$$

**Distribuția exponențială.** Pentru lucrarea de față prezintă interes zona B a caracteristicii, care corespunde vieții utile a unei componente a unui sistem. Subliniez însă faptul că, în funcție de modul în care sunt interconectate componentele sistemului pentru a asigura funcționalitatea acestuia, nu va rezulta neapărat o funcție de hazard constantă și pentru sistem. Pentru detalii se recomandă [GROSH89]. Rata de defectare constantă se obține pentru cazul particular când parametrul distribuției Weibull se ia  $\beta = 1$ . În acest caz, pe baza relațiilor precedente, rezultă:

$$h(t) = \lambda = \text{constant} \quad (3.4.4)$$

$$F(t) = 1 - e^{-\lambda t} \quad (3.4.5)$$

$$R(t) = e^{-\lambda t} \quad (3.4.6)$$

$$f(t) = \lambda e^{-\lambda t} \quad (3.4.7)$$

Legea de repartiție caracterizată prin funcțiile (3.4.5-3.4.7) constituie o *distribuție (negativ) exponențială* (negative exponential distribution), cea mai frecvent utilizată distribuție de tip continuu în teoria fiabilității, fiind singura care prezintă o rată de defectare constantă. Timpul mediu până la defectare (*MTTF*) se calculează conform (3.2.13) ca fiind:

$$MTTF = E[T] = \int_0^{\infty} e^{-\lambda t} dt = \frac{1}{\lambda} \quad (3.4.8)$$

Distribuția exponențială este adecvată pentru a modela comportarea componentelor (inclusiv cele electronice), care dacă sunt bune se consideră ca și cum ar fi noi (used-good-as-new). Acest lucru se datorează proprietății de a nu ține cont de evoluția trecută, ceea ce rezultă din următoarea teoremă [GROSH89]:

**Teorema 3.4.1.** În cazul unei distribuții exponențiale a timpului până la defectare,  $F(t) = 1 - e^{-\lambda t}$ , ( $t \geq 0$ ), timpul așteptat până la următoarea defectare este  $1/\lambda$  indiferent de momentul în care se începe observarea.

*Demonstrație.* Conform ipotezei,  $F(t) = \Pr(T \leq t) = 1 - e^{-\lambda t}$ . Pornind observarea de la  $t=0$ , timpul mediu până la defectare este  $1/\lambda$ , conform (3.4.8). Presupunând că până la noul moment de observare  $t = x$  nu a apărut defectarea, distribuția timpului rămas până la defectare este  $\Pr(T-x \leq t \mid T > x)$ . Aceasta este o probabilitate condiționată de faptul că defectarea să nu fi apărut până în acest moment, care se calculează cu formula  $\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr B}$ :

$$\begin{aligned}\Pr(T-x \leq t | T > x) &= \frac{\Pr(x < T \leq x+t)}{\Pr(T > x)} = \frac{\Pr(T \leq x+t) - \Pr(T < x)}{\Pr(T > x)} = \\ &= \frac{(1 - e^{-\lambda(x+t)}) - (1 - e^{-\lambda x})}{e^{-\lambda x}} = \frac{e^{-\lambda x} (1 - e^{-\lambda t})}{e^{-\lambda x}} = 1 - e^{-\lambda t} = F(t)\end{aligned}$$

S-a obținut aceeași distribuție ca pentru situația  $t=0$ , media variabilei aleatoare fiind și aici  $1/\lambda$ .

□

Datorită acestei proprietăți de a fi fără memorarea trecutului (memoryless property), în perioada duratei de viață cu rata constantă a defectărilor, componentele se defectează nu din cauza uzurii, ci din cauza unor influențe externe care apar într-o manieră Poisson [GROSH89]. De exemplu un microprocesor va cădea datorită unui supracurent și nu a uzurii. Asemenea defectări sunt accidentale, aleatoare, corespunzătoare unei funcționări normale. Un proces Poisson creează puncte regenerative, proprietate exploatată de programele de evaluare a fiabilității și performanțelor prin simulare [CUTR94][LIN93][VĂDU77].

**Distribuția geometrică.** Legea de repartiție de tip discret similară celei exponențiale, este *distribuția geometrică* (geometric distribution). Aceasta prezintă de asemenea proprietatea de a fi fără memorarea trecutului, așa cum se va demonstra mai jos.

Distribuția geometrică caracterizează numărul de încercări de tip Bernoulli până la prima reușită (inclusiv aceasta). O încercare Bernoulli este aceea al cărei rezultat nu poate fi decât reușit sau nereușit. În cazul de față folosesc această distribuție pentru descrierea matematică a timpului de așteptare (de tip discret  $t=1,2,3, \dots$ ) până la apariția defectului. Parametrul distribuției este probabilitatea de reușită dintr-o încercare (acestea fiind identice), cu alte cuvinte probabilitatea de defectare în "încercarea" de la oricare moment  $t$ , notată cu  $q$  ( $0 \leq q \leq 1$ ). Prezintă interes expresia funcției de masă de probabilitate, adică a probabilității ca defectarea să apară la momentul  $t$ ,  $f(t) = \Pr(T=t)$ . Raționamentul de calcul este următorul:

- probabilitatea de defectare la timpul  $T = 1$  este:  $q$
- probabilitatea de defectare la timpul  $T = 2$ , astfel încât componenta a funcționat în momentele anterioare, este:  $q(1-q)$
- probabilitatea de defectare la timpul  $T = t$ , astfel încât componenta a funcționat în momentele anterioare, este:  $q(1-q)^{t-1}$

Rezultă deci formula funcției de masă a distribuției geometrice, o funcție monoton descrescătoare, ca fiind:

$$f(t) = q(1-q)^{t-1} \quad (3.4.9)$$

Funcția de distribuție cumulată se obține pe baza relației (3.3.3) ca suma unei progresii geometrice cu rata  $1-q$ :

$$F(t) = \sum_{i=1}^t q(1-q)^{i-1} = q + q(1-q) + q(1-q)^2 + \dots + q(1-q)^{t-1} = 1 - (1-q)^t \quad (3.4.10)$$

Pe baza complementarității, funcția de fiabilitate este:

$$R(t) = 1 - F(t) = (1-q)^t \quad (3.4.11)$$

Rezultatul apare ca și normal, dacă se are în vedere faptul că funcția de fiabilitate exprimă probabilitatea ca produsul să funcționeze fără defectare pe durata a  $t$  unități (discrete) de timp.

Numărul de unități de timp până la defectare se calculează cu relația (3.3.6), descompunând expresia în mai multe progresii geometrice infinite cu rata subunitară:

$$\begin{aligned} MTTF = E[T] &= \sum_{i=1}^{\infty} i q (1-q)^{i-1} = q + 2q(1-q) + 3q(1-q)^2 + 4q(1-q)^3 + \dots = \\ &= q + q(1-q) + q(1-q)^2 + q(1-q)^3 + \dots + \\ &\quad + q(1-q) + q(1-q)^2 + q(1-q)^3 + \dots + \\ &\quad + q(1-q)^2 + q(1-q)^3 + \dots + \\ &\quad \dots \dots \dots = \\ &= 1 + (1-q) + (1-q)^2 + \dots = \frac{1}{q} \end{aligned} \quad (3.4.12)$$

Distribuția geometrică este singura distribuție discretă fără memorarea trecutului (memoryless/forgetfulness distribution), adică reținerea rezultatelor "încercărilor" trecute nu este semnificativă în prezicerea comportării viitoare.

**Teorema 3.4.2.** Fiind dată o distribuție geometrică a timpului (discret) până la defectare,  $F(t) = 1 - (1-q)^t$ , ( $t=1, 2, 3, \dots$ ), timpul mediu până la defectare este același ( $1/q$ ), indiferent de momentul în care se începe observarea.

*Demonstrație.* Demonstrația urmează același raționament ca la teorema 3.4.1. Probabilitatea ca defectarea să apară până în momentul  $x+t$ , condiționată de faptul că până la momentul  $x$  ea nu s-a produs, este dată de:

$$\begin{aligned} \Pr(T \leq x+t | T > x) &= \frac{\Pr(x < T \leq x+t)}{\Pr(T > x)} = \frac{\Pr(T \leq x+T) - \Pr(T \leq x)}{\Pr(T > x)} = \\ &= \frac{[1 - (1-q)^{x+t}] - [1 - (1-q)^x]}{(1-q)^x} = \frac{(1-q)^x [1 - (1-q)^{x+t}]}{(1-q)^x} = \\ &= 1 - (1-q)^t = \Pr(T \leq t) \end{aligned}$$

Expresia funcției de distribuție  $F(t) = \Pr(T \leq t)$  fiind aceeași indiferent dacă observarea ("măsurarea" timpului) începe la momentul  $t=0$  sau  $t=x$ , rezultă că și valoarea lui  $E[T]$  este aceeași în ambele situații.

În finalul acestui paragraf prezint în tabelul 3.4.1 o comparație între distribuția exponențială (continuă) și cea geometrică (discretă), pentru a pune în evidență

similitudinile dintre ele. Folosind notația  $p=1-q$  se observă corespondența  $p=e^{-\lambda}$ , în particular pentru orice  $0 \leq p \leq 1$  se poate găsi un  $0 \leq \lambda \leq \infty$  care să satisfacă această egalitate. Și încă o constatare interesantă: la distribuția exponențială rata de defectare este dată de  $\lambda=\text{constant}$ , iar la distribuția geometrică această rată (care exprimă probabilitatea de defectare pe unitatea de timp la momentul  $t$ , cu condiția ca defectul să nu fi apărut până la  $t-1$  inclusiv) este dată tocmai de  $q=\text{constant}$ , deoarece:

$$\begin{aligned} \Pr(T = t | T > t - 1) &= \frac{\Pr(t - 1 < T \leq t)}{\Pr(T > t - 1)} = \frac{\Pr(T = t)}{\Pr(T > t - 1)} = \\ &= \frac{f(t)}{R(t-1)} = \frac{q(1-q)^{t-1}}{(1-q)^{t-1}} = q \end{aligned} \quad (3.4.13)$$

Distribuția geometrică poate fi considerată un caz particular al distribuției Pascal, care exprimă numărul de încercări (aici unități de timp) până la a  $m$ -a realizare, inclusiv aceasta. Luând  $m=1$  se obține distribuția geometrică. Distribuția Pascal se folosește pentru a modela numărul de biți necesari pentru a transmite cu succes  $m$  biți utili [JAIN91].

Criteriul	Distribuția exponențială	Distribuția geometrică
Variabila aleatoare	Continuă $t \geq 0$	Discretă $t = 1, 2, 3, \dots$
Parametrul	$\lambda = \text{constant}$	$q = \text{constant}$
Funcția de densitate/masă	$\lambda e^{-\lambda t} = \lambda(e^{-\lambda})^t$	$qp^{t-1} = (q/p)p^t$
Funcția de distribuție $\Pr(T \leq t)$	$1 - e^{-\lambda t} = 1 - (e^{-\lambda})^t$	$1 - p^t$
Media $\mu$	$1/\lambda$	$1/q$
Dispersia $\sigma^2$	$1/\lambda^2$	$p/q^2$

Tabelul 3.4.1.

### 3.5. Funcția de structură a sistemelor bivalente

Problema centrală a teoriei fiabilității constă în studiul fiabilității sistemelor pe baza fiabilității componentelor. Pentru aceasta sunt necesare modele matematice care captează simultan relațiile logice și structurale dintre componentele sistemului, precum și evenimentele aleatoare de tipul defectărilor componentelor. Un astfel de model, general valabil, este descrierea sistemului cu funcția de structură. Pentru a o defini, trebuie formulate în prealabil următoarele ipoteze:

1. Sistemul are numai două stări posibile: funcțional sau defect.
2. Sistemul poate fi descompus în  $n$  componente, astfel încât la un moment dat fiecare componentă se poate afla într-una din stările funcțională sau defectă. Starea sistemului depinde numai de stările componentelor sale.
3. Fiecare componentă  $c_i$  ( $i = 1, 2, \dots, n$ ) a sistemului posedă o durată de viață aleatoare  $T_i$ , căreia îi corespunde câte o funcție de fiabilitate  $R_i(t)$ :

$$R_i(t) = \Pr(T_i > t) \quad (i = 1, 2, \dots, n)$$

4. Variabilele aleatoare  $T_i$  sunt independente.

Sistemul astfel considerat, în care atât componentele, cât și sistemul pot avea numai două stări, este un sistem bivalent. La un moment dat  $t$ , o componentă  $c_i$  are probabilitatea  $p_i$  de a fi în stare funcțională, respectiv probabilitatea  $q_i$  de a fi defectă, astfel încât:

$$p_i = R_i(t) \quad (3.5.1)$$

$$q_i = 1 - p_i = F_i(t) \quad (3.5.2)$$

În relația precedentă  $F_i(t)$  reprezintă funcția de distribuție cumulată a timpului până la defectare pentru componenta  $c_i$ , conform unei anumite legi de distribuție. Fiecărei componente  $i$  se asociază o variabilă aleatoare Bernoulli, adică o variabilă binară  $x_i$ , care reprezintă starea componentei  $c_i$  la un moment  $t$ , astfel încât prin convenție:

$$x_i = \begin{cases} 1 & \text{dacă } c_i \text{ este funcțională} \\ 0 & \text{dacă } c_i \text{ este defectă} \end{cases}$$

Rezultă imediat că:

$$p_i = \Pr(c_i \text{ este bun}) = \Pr(x_i=1) = E[x_i] \quad (3.5.3)$$

**Definiția 3.5.1.** Se numește *funcție de structură* (structure function) a unui sistem bivalent cu  $n$  componente, o funcție  $\Phi : \{0,1\}^n \rightarrow \{0,1\}$  indicatoare a stării sistemului în funcție de stările componentelor sale, astfel încât:

$$\Phi(x_1, x_2, \dots, x_n) = \begin{cases} 1 & \text{dacă sistemul este funcțional} \\ 0 & \text{dacă sistemul este defect} \end{cases}$$

Relația dintre funcția de structură și fiabilitatea sistemului este dată de:

$$R = \Pr(\text{sistemul este bun}) = \Pr(\Phi=1) = E[\Phi] \quad (3.5.4)$$

Cu alte cuvinte funcția de fiabilitate a unui sistem este egală cu valoarea așteptată (media) funcției sale aleatoare de structură.

Subliniez faptul că funcția de structură a unui sistem bivalent poate fi exprimată în funcție de variabilele sale, atât sub formă booleană cât și algebrică. Acest lucru este posibil, deoarece pe mulțimea  $\{0,1\}$  se consideră atât structura de algebră Boole, cât și cea de corp comutativ [TĂRC89]. Pentru a evita confuziile ulterioare, voi ilustra cu un exemplu faptul că funcția de fiabilitate se poate exprima pornind de la ambele reprezentări ale funcției de structură.

**Exemplul 3.5.2.** Fie sistemul cu trei componente, reprezentat prin schema bloc de fiabilitate din figura 3.5.1. Condiția ca sistemul să fie operațional este ca să funcționeze componenta  $c_1$ , precum și una din componentele  $c_2$  sau  $c_3$ . Corespunzător, definiția tabelară a funcției de structură  $\Phi$  este cea din tabelul 3.5.2. Considerând pe  $\Phi$  ca o funcție booleană, din tabel se obține forma ei disjunctivă:

$$\Phi = x_1 \bar{x}_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3 = x_1 x_2 \vee x_1 x_3$$



(Pentru simplitatea scrierii, fără a afecta însă înțelegerea, am omis operatorul  $\cap$ , deci în loc de  $x_1 \bar{x}_2 x_3$  se va subînțelege  $x_1 \cap \bar{x}_2 \cap x_3$  ș.a.m.d.). Aceeași funcție se poate exprima și sub formă algebrică, folosind operațiile de adunare și înmulțire:

$$\Phi = x_1[1 - (1 - x_2)(1 - x_3)] = x_1 x_2 + x_1 x_3 - x_1 x_2 x_3$$

□

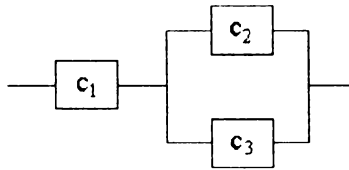


Figura 3.5.1.

$x_1$	$x_2$	$x_3$	$\Phi$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Tabelul 3.5.2.

Întrucât evaluarea lui  $R$  pe baza expresiei lui  $\Phi$  necesită conform relației (3.5.4) calcule probabilistice, voi trece în revistă câteva formule de bază din teoria probabilităților [TÂRC89] [BENN82] [JAIN91]:

Folosesc operatorul  $\cup$  pentru relația SAU, respectiv  $\cap$  pentru ȘI.

Fiind dat un eveniment  $A$  și complementul său  $\bar{A}$ , există egalitățile:

$$\Pr(A \cup \bar{A}) = 1 \quad (3.5.5)$$

$$\Pr(\bar{A}) = 1 - \Pr(A) \quad (3.5.6)$$

Pentru două evenimente  $A$  și  $B$ :

$$\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B) \quad (3.5.7)$$

Dacă  $A$  și  $B$  sunt independente, atunci:

$$\Pr(A \cap B) = \Pr(A) \cdot \Pr(B) \quad (3.5.8)$$

Dacă  $A$  și  $B$  nu se pot produce simultan, adică sunt incompatibile (sau reciproc exclusive, sau disjuncte), atunci:

$$\Pr(A \cap B) = 0 \quad (3.5.9)$$

În cazul general, când  $A$  și  $B$  sunt dependente:

$$\Pr(A \cap B) = \Pr(A) \cdot \Pr(B | A) = \Pr(B) \cdot \Pr(A | B) \quad (3.5.10)$$

unde  $A | B$  reprezintă producerea evenimentului  $A$  condiționată de producerea lui  $B$ . Probabilitatea acestuia se va exprima deci ca:

$$\Pr(A | B) = \frac{\Pr(A \cap B)}{\Pr(B)} \quad (3.5.11)$$

Probabilitatea evenimentului  $A$  exprimată în funcție de probabilitatea unei cauze  $B$  se obține cu formula lui Bayes:

$$\Pr(A) = \Pr(A|B) \cdot \Pr(B) + \Pr(A|\bar{B}) \cdot \Pr(\bar{B}) \quad (3.5.12)$$

Dacă  $X, Y$  reprezintă două variabile aleatoare și  $a, b$  două constante, atunci are loc egalitatea (liniaritatea operatorului  $E$ ):

$$E[aX \pm bY] = aE[X] \pm bE[Y] \quad (3.5.13)$$

Pentru variabile independente:

$$E[XY] = E[X] \cdot E[Y] \quad (3.5.14)$$

Relațiile logice dintre componentele unui sistem din punct de vedere al fiabilității pot fi puse în evidență cel mai simplu cu ajutorul schemelor bloc de fiabilitate [VLĂD82]. Componentele a căror funcționare este strict necesară pentru ca sistemul să fie operațional sunt conectate în serie, iar componentele redundante, care se pot substitui reciproc, sunt conectate în paralel pe aceste scheme. Subliniez că aceste scheme bloc nu reprezintă conexiunile fizice ale componentelor, ci numai relațiile dintre ele pentru o funcționare cu succes a sistemului.

În cazul unui sistem serie (serial system) cu  $n$  componente (figura 3.5.3) având vectorul de stare  $X = (x_1, x_2, \dots, x_n)$ , funcția de structură sub formă algebrică are expresia:

$$\Phi(X) = x_1 x_2 \dots x_n = \min(x_1, x_2, \dots, x_n) \quad (3.5.15)$$

Funcția de fiabilitate corespunzătoare este, folosind (3.5.4), (3.5.14), (3.5.3) și având în vedere independența componentelor:

$$R = E[x_1 x_2 \dots x_n] = E[x_1] E[x_2] \dots E[x_n] = p_1 p_2 \dots p_n \quad (3.5.16)$$

Același rezultat se obține dacă funcția de structură se exprimă sub formă booleană:

$$\Phi(X) = x_1 \wedge x_2 \wedge \dots \wedge x_n \quad (3.5.17)$$

Folosind de data aceasta relația (3.5.8) în loc de (3.5.14), rezultă:

$$R = \Pr(x_1 \wedge x_2 \wedge \dots \wedge x_n = 1) = \Pr(x_1 = 1) \Pr(x_2 = 1) \dots \Pr(x_n = 1) = p_1 p_2 \dots p_n \quad (3.5.18)$$

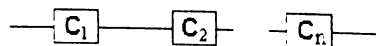


Figura 3.5.3.

În cazul unui sistem paralel (parallel system) (figura 3.5.4), funcția de structură sub cele două forme se exprimă astfel:

$$\Phi(X) = 1 - (1 - x_1)(1 - x_2) \dots (1 - x_n) = \max(x_1, x_2, \dots, x_n) \quad (3.5.19)$$

$$\Phi(x) = x_1 \vee x_2 \vee \dots \vee x_n \quad (3.5.20)$$

Funcția de fiabilitate, conform celor două forme de exprimare și ținând cont de independența componentelor, va fi dată pe baza (3.5.3), (3.5.4), (3.5.13), (3.5.14) de

$$R = E[1 - (1-x_1)(1-x_2) \cdots (1-x_n)] = 1 - E[(1-x_1)(1-x_2) \cdots (1-x_n)] = \\ = 1 - \{1-E[x_1]\} \{1-E[x_2]\} \cdots \{1-E[x_n]\} = 1 - (1-p_1)(1-p_2) \cdots (1-p_n) \quad (3.5.21)$$

respectiv:

$$R = \Pr(x_1 \cup x_2 \cup \dots \cup x_n) = 1 - \overline{\Pr(x_1 \cup x_2 \cup \dots \cup x_n = 1)} = \\ = 1 - \Pr(\bar{x}_1 \cap \bar{x}_2 \cap \dots \cap \bar{x}_n = 1) = 1 - \Pr(\bar{x}_1 = 1) \cdot \Pr(\bar{x}_2 = 1) \cdots \Pr(\bar{x}_n = 1) = \\ = 1 - [1 - \Pr(x_1 = 1)][1 - \Pr(x_2 = 1)] \cdots [1 - \Pr(x_n = 1)] = \\ = 1 - (1-p_1)(1-p_2) \cdots (1-p_n) \quad (3.5.22)$$

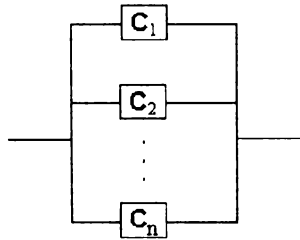


Figura 3.5.4.

Evident prin cele două metode de calcul s-a obținut aceeași expresie pentru  $R$ .

În cazul sistemelor  $k$ -din- $n$  ( $k$ -out-of- $n$ -system), sistemul se consideră operațional dacă cel puțin  $k$  din cele  $n$  componente sunt funcționale. Evaluarea fiabilității acestor sisteme, mult mai generale decât cele pur seriale sau paralele, este importantă deoarece execuția unui task într-un sistem distribuit poate presupune un minim de  $k$  noduri active pentru prelucrare [SHAN82][RAI87]. Funcția de structură se definește astfel:

$$\Phi(Y) = \begin{cases} 1 & \text{dacă } \sum_{i=1}^n x_i \geq k \\ 0 & \text{dacă } \sum_{i=1}^n x_i < k \end{cases} \quad (3.5.23)$$

Calculul funcției de fiabilitate se ilustrează pe un exemplu.

**Exemplul 3.5.3.** Se consideră un sistem cu redundanță modulară triplă (TMR, triple modular redundancy), de tipul 2-din-3. Schema bloc de fiabilitate este dată în figura 3.5.5. Folosind relațiile (3.5.15) și (3.5.19), funcția de structură (algebrică) este:

$$\Phi = 1 - (1-x_1x_2)(1-x_1x_3)(1-x_2x_3)$$

Utilizând proprietatea de idempotență a unei variabile binare ( $x_i x_i = 1$ ), prin prelucrări succesive se obține:

$$\Phi = x_1x_2 + x_1x_3 + x_2x_3 - 2x_1x_2x_3$$

Rezultă imediat funcția de fiabilitate, ca fiind:

$$R = E[\Phi] = p_1p_2 + p_1p_3 + p_2p_3 - 2p_1p_2p_3$$

Deoarece fiecare componentă apare de două ori în schema de fiabilitate, dacă se lucrează cu funcția de structură booleană trebuie ținut cont de faptul că evenimentele asociate termenilor funcției nu sunt independente (formula (3.5.7)):

$$\begin{aligned} R &= \Pr[x_1x_2 \cup x_1x_3 \cup x_2x_3 = 1] = \Pr(x_1x_2 \cup x_1x_3 = 1) + \Pr(x_2x_3 = 1) - \\ &\quad - \Pr[(x_1x_2 \cup x_1x_3)x_2x_3 = 1] = \Pr(x_1x_2 = 1) + \Pr(x_1x_3 = 1) - \Pr(x_1x_2 = 1) - \\ &\quad - \Pr(x_1x_2x_3 \cup x_1x_2x_3 = 1) = \Pr(x_1x_2 = 1) + \Pr(x_1x_3 = 1) + \Pr(x_2x_3 = 1) - \\ &\quad - 2\Pr(x_1x_2x_3 = 1) = p_1p_2 + p_1p_3 + p_2p_3 - 2p_1p_2p_3 \end{aligned}$$

□

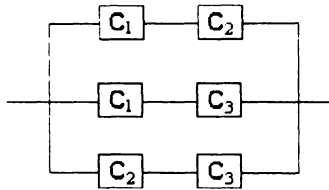


Figura 3.5.5.

Un sistem cu  $n$  componente implică existența a  $2^n$  vectori care formează mulțimea stărilor componentelor. Între doi vectori de stare  $X = (x_1, x_2, \dots, x_n)$  și  $Y = (y_1, y_2, \dots, y_n)$  se poate defini o relație de ordine astfel:

$$\begin{aligned} X &= Y && \text{dacă și numai dacă } x_i = y_i \quad \forall i = 1, 2, \dots, n \\ X &\geq Y && \text{dacă și numai dacă } x_i \geq y_i \quad \forall i = 1, 2, \dots, n \\ X &> Y && \text{dacă și numai dacă } x_i > y_i \quad \forall i = 1, 2, \dots, n \end{aligned}$$

**Definiția 3.5.4.** Un sistem cu funcția de structură  $\Phi$  este *monoton* sau *coerent* (coherent system) dacă pentru orice pereche de vectori de stare satisface proprietatea că  $X \geq Y$  implică:

$$\Phi(X) \geq \Phi(Y) \quad (3.5.24)$$

Pentru ca un sistem să fie coerent, trebuie să fie îndeplinite următoarele două condiții [VISW87]:

1.  $\Phi(X)$  este nedescrescătoare pentru fiecare din variabilele sale.
2. Fiecare componentă este *relevantă*, adică pentru orice  $i$  există o combinație  $(x_1, x_2, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$  pentru care este adevărată relația:

$$\Phi(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n) > \Phi(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \quad (3.5.25)$$

Cu alte cuvinte, un sistem este coerent dacă defectarea oricăreia din componentele sale nu poate duce la restabilirea stării funcționale a sistemului. O componentă  $c_j$  este *irrelevantă* (inutilă) din punct de vedere al funcționării sistemului dacă pentru orice  $x_j$  ( $j \neq i$ ):

$$\Phi(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n) = \Phi(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \quad (3.5.26)$$

Ca o completare, menționez că în afara schemei bloc de fiabilitate folosită mai înainte, legăturile între componentele unui sistem din punct de vedere fiabilistic pot fi reprezentate și cu ajutorul grafului (rețelei) de fiabilitate sau al arborelui de defectare [TÂRC89][VISW87][HOHAN82][GEBER84][WOOD85]. Graful de fiabilitate este un graf orientat fără bucle în care se disting două noduri (sursă și terminal) și în care fiecare arc corespunde unei componente (unei componente îi pot corespunde însă și mai multe arce). Arborele de defectare este o schemă logică care arată modurile în care defectările componentelor se pot combina pentru a duce la defectarea sistemului. Dacă sistemul are mai multe condiții de avarie, atunci pentru fiecare trebuie construit câte un arbore separat. În asemenea situații este mai indicată folosirea modelelor Markov, greu de manipulat însă în sistemele cu multe componente și deci multe stări [SHAR93].

### 3.6. Sisteme de tip serie-paralel

În general un sistem serie-paralel este acela a cărui schemă bloc de fiabilitate este compusă din cele două structuri de bază: de tip serial și paralel. Rezolvarea unui astfel de sistem, adică determinarea fiabilității sale în funcție de fiabilitățile componentelor, se face prin *metoda descompunerii* (decomposition). Ideea de bază este de a identifica și reuni în etape succesive structurile fundamentale (serie și paralel), aplicând relațiile (3.5.18) și (3.5.22) cu condiția ca defectările componentelor să fie independente, până când schema se reduce la un singur bloc.

În diverse domenii (comunicații, rețele de calculatoare) este necesară determinarea fiabilității unui sistem modelat printr-un graf probabilistic, care redă topologia rețelei. Fie acesta un graf neorientat  $G = (V, E)$ , ale cărui muchii din  $E$  se pot defecta independent cu probabilități cunoscute și în care  $K \subseteq V$  este o submulțime de noduri. Problema de analiză a fiabilității constă în determinarea probabilității ca toate nodurile din  $K$  să rămână conexe (conectate), această mărime fiind numită *fiabilitate de tip K-terminal* (*K-terminal reliability*) a rețelei, notată  $R(G_K)$ . Ca și cazuri particulare, pentru  $|K| = 2$  se obține problema conectivității unei perechi de terminale (*2-terminal reliability*), iar pentru  $K=V$  problema fiabilității globale (*all-terminal reliability*) a rețelei (identică cu problema conectivității din teoria grafurilor).

Calculul mărimii  $R(G_K)$  reprezintă în general o problemă complexă, de aceea se recurge la procedee de reducere a dimensiunii grafului cu menținerea fiabilității (*reliability preserving reductions*).

**Definiția 3.6.1.** Un graf probabilistic care prin substituiri succesive de tip serial și paralel poate fi redus la o singură muchie este un *graf serie-paralel*.

Pentru determinarea fiabilității de tip *K-terminal* este însă necesar ca în urma reducerii cele două noduri rămase să facă parte din  $K$ . Se spune în acest caz că graful  $G_K$  este *reductibil*, un exemplu fiind dat în figura 3.6.1.a pentru  $K = \{v_2, v_3, v_3\}$  pe o structură care este un graf modificat al rețelei ARPA [RAI78][RAI86a]. Dacă însă  $K = \{v_1, v_6\}$ , graful este ireductibil prin procedeele simple de reducere (figura 3.6.1.b). Deci, în funcție

de nodurile care fac parte din  $K$  și evident topologia lui  $G$ , o reducere simplă a acestuia cu menținerea fiabilității poate fi sau nu posibilă.

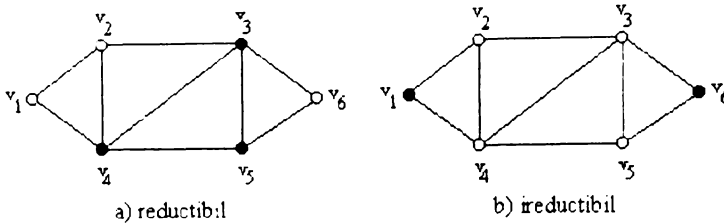


Figura 3.6.1.

**Reducțiile simple.** Reducțiile simple uzuale sunt cele de tip *paralel*, *serial* și de *gradul 2*, ultima fiind de fapt tot o reducere serială. Un caz aparte îl reprezintă *transformarea pendinte* (pendant transformation), când muchia “atârână” de restul grafului. În tabelul 3.6.2 se prezintă o sinteză a acestor procedee, în cazul în care nodurile sunt perfecte, iar o muchie  $e_i$  are fiabilitatea  $p_i$  și probabilitatea de defectare  $q_i = 1 - p_i$ . O astfel de reducere trebuie să ducă la: eliminarea anumitor muchii și/sau noduri din  $G$ , obținându-se  $G'$ : definirea noilor fiabilități de muchii; definirea noului set  $K'$ ; definirea unui factor de multiplicare  $\Omega$  astfel încât:

$$R(G_K) = \Omega R(G'_K) \quad (3.6.1)$$

Se remarcă în cazul reducerii de gradul 2 (degree-2 reduction), când nodul eliminat face parte din  $K$ , a unui factor  $\Omega=1$ . Este clar că pentru problema fiabilității globale ( $K=I$ ), orice graf serial-paralel  $G$  este reductibil, deoarece fiecare nod de gradul 2 din  $G$  permite o reducere serială de gradul 2, cele două noduri adiacente fiind oricum din  $K$ . Deci și cele două noduri finale vor face parte din  $K$ . În cazul fiabilității de tip  $K$ -terminal acest lucru nu este totdeauna posibil și depinde de aranjamentul nodurilor țintă (din  $K$ ). Fie  $v$  un nod de gradul 2, adiacent cu  $u$  și  $w$ . Există următoarele situații [SHOO92]:

1. Dacă  $v \notin K$ , se poate utiliza transformarea serială clasică, deoarece nu are importanță ca acest nod să poată neapărat comunica cu celelalte. Cele două muchii incidente la  $v$  se înlocuiesc cu una singură având fiabilitatea  $p_1 p_2$ .
2. Dacă  $u, v, w \in K$ , nodul  $v$  se poate elimina din  $G_K$ , muchiile incidente fiind înlocuite cu una singură printr-o transformare serială de gradul 2. Fiabilitatea noului graf  $G'_K$  trebuie condiționată de faptul ca  $v$  să poată comunica cu cel puțin unul din vecini. Probabilitatea acestui eveniment este dată de faptul ca una din cele două muchii să fie operațională, adică  $\Omega = 1 - (1 - p_1)(1 - p_2)$ . Notând cu  $A$  evenimentul “ $u$  comunică cu  $w$ ”, iar cu  $B$  evenimentul “ $v$  comunică cu  $u$  sau  $w$ ”, pe baza relației de definiție a probabilității condiționate (3.5.11) se obține:

$$p_{12} = \Pr(A \mid B) = \frac{\Pr(A \cap B)}{\Pr(B)} = \frac{p_1 p_2}{1 - (1 - p_1)(1 - p_2)} \quad (3.6.2)$$

De factorul de multiplicare  $\Omega$  trebuie ținut cont atât la fiabilitatea noii muchii,  $p_{12} = p_1 p_2 / \Omega$ , cât și la calculul lui  $R(G)$  conform (3.6.1).

3. În celelalte trei cazuri ( $v \in K$  și unul din  $u, w$  sau ambele nu fac parte din  $K$ ) nu se poate face o transformare simplă, deoarece nu mai este clară probabilitatea pe care să o folosim pentru evenimentul ca  $v$  să poată comunica cu restul rețelei. Graful serie-parallel care prezintă o astfel de situație este considerat ireductibil.

Paralel	Serial ( $v \notin K$ )	Gradul 2 ( $u, v, w \in K$ )
$p_{12} = 1 - q_1 q_2$ $K' = K$ $\Omega = 1$	$p_{12} = p_1 p_2$ $K' = K$ $\Omega = 1$	$p_{12} = \frac{p_1 p_2}{1 - q_1 q_2}$ $K' = K - v$ $\Omega = 1 - q_1 q_2$
Pendinte ( $v \notin K$ )	Pendinte ( $v \in K$ )	Legenda
		<ul style="list-style-type: none"> <li>● nod țintă (target), din <math>K</math></li> <li>○ nodul nu este nod țintă (non-target), din <math>V - K</math></li> <li>⊗ nod oarecare, din <math>V</math></li> </ul>
$K' = K$ $\Omega = 1$	$K' = K - u - v + w$ $\Omega = p_1$	

Tabelul 3.6.2.

În continuare, ca o contribuție personală, voi analiza ce modificări implică asupra regulilor de reducere din tabelul 3.6.2 situația când modelarea rețelei se face printr-un graf orientat  $D = (V, E)$ . În acest caz muchia corespunde unei legături unidirecționale între nodurile vecine. Fiabilitatea sistemului se calculează de regulă raportată la un nod sursă  $s \in V$ , indicatorul corespunzător fiind fiabilitatea de tip *sursă-la-K-terminale* (SKT, source-to-K-terminal). Aceasta, notată  $R(D_K)$ , exprimă probabilitatea ca să existe cel puțin un drum orientat de la  $s$  la fiecare nod din  $K \subseteq V$ . Dacă  $K = V$ , atunci fiabilitatea este de tip *sursă-la-toate-terminale* (SAT, source-to-all-terminal), iar pentru  $|K|=2$ , problema devine o fiabilitate de tip *sursă-la-terminal* (ST, source-to-terminal) în  $G$ .

În ceea ce privește utilizarea unor reducere simple de tip serial, paralel și de gradul 2, ca o primă constatare se remarcă că acestea sunt aplicabile numai dacă muchiile  $e_1$  și  $e_2$  au aceeași orientare. Rezultă simplu că pentru transformarea paralelă, serială și pendinte regulile de la grafuri neorientate rămân valabile. Este de notat faptul că dacă există în  $D$

un arc originar într-un nod țintă pendinte  $v \neq s$ , atunci  $R(D_K) = 0$ . Arcul aferent unui nod pendinte  $v \notin K$  este irelevant și poate fi eliminat. De asemenea sunt irelevante arcele incidente la un nod  $v \notin K$  dacă toate au același sens raportat la acest nod. În cazul unui nod  $v$  de gradul 2 există două posibilități:

1.  $v$  nu este nod sursă ( $v \neq s$ ). În această situație probabilitatea "u comunică cu w" trebuie condiționată numai de "u comunică cu v", deoarece dacă  $e_1$  se defectează,  $v$  rămâne izolat datorită sensului lui  $e_2$ . Rezultă  $p_{12} = \frac{p_1 p_2}{p_1} = p_2$  și  $\Omega = p_1$ . Este important faptul că dacă  $u \in K$  în  $D$ , atunci în  $D'$  se va înlocui cu  $t \in K$ , pentru a se asigura conectarea lui  $v$  la restul grafului.
2.  $v$  este nod-sursă ( $v = s$ ). În această situație este suficient să fie adevărat evenimentul "v comunică cu w" și ca urmare  $R(D_K) = p_2 R(D'_K)$ , arcul  $e_2$  fiind eliminat. De asemenea se poate elimina și  $e_1$ , care este inutil, pe baza lemei următoare:

**Lema 3.6.2.** Într-un graf probabilistic orientat  $D = (V, E)$  un arc incident spre interior în nodul sursă este irelevant.

*Demonstrație.* Rezultă imediat că afirmația este adevărată, dacă se are în vedere că arcul în cauză nu mai conectează un nou nod din  $K$  la restul grafului. Fiabilitatea grafului lipsit de acest arc, raportată la  $K$ , este identică cu cea a grafului inițial  $R(D_K|e) = R(D_K)$ .  $\square$

Tabelul 3.6.3 prezintă o sinteză a transformărilor posibile în cazul muchiilor orientate. Se remarcă faptul că în acest caz, spre deosebire de cazul grafului neorientat, reducția de gradul 2 se poate face și în situația în care  $u, w \notin K$ . Voi ilustra cele prezentate cu un exemplu.

**Exemplul 3.6.3.** În sistemul din figura 3.6.4 nodul 1 este nodul sursă. Subliniez că o condiție a reductibilității este ca nodurile finale (aici 1 și 5) să fie din  $K$ . Fiabilitatea raportată la mulțimea de noduri  $\{1, 2, 5\}$  se determină prin parcurgerea următorilor pași:

- se elimină  $e_6$  și  $e_7$ , fiind irelevante (cazul 5)

- se compune  $e_1$  în serie cu  $e_2$  (cazul 3):

$$p_{12} = p_2$$

$$\Omega = p_1$$

- se compune  $e_{12}$  în paralel cu  $e_3$  (cazul 1):

$$p_{123} = 1 - (1 - p_2)(1 - p_3) = p_2 + p_3 - p_2 p_3$$

- se compune  $e_{123}$  în serie cu  $e_5$  (cazul 2):

$$p_{1235} = p_{123} p_5 = p_2 p_5 + p_3 p_5 - p_2 p_3 p_5$$

- se compune  $e_{1235}$  în paralel cu  $e_4$  (cazul 1):

$$p_{12345} = 1 - (1 - p_{1235})(1 - p_4) = p_4 + p_2 p_5 + p_3 p_5 - p_2 p_3 p_5 - p_2 p_4 p_5 - p_3 p_4 p_5 + p_2 p_3 p_4 p_5$$

- fiabilitatea sistemului (formula (3.6.1)):

$$R = \Omega p_{12345} = p_1 p_4 + p_1 p_2 p_5 + p_1 p_3 p_5 - p_1 p_2 p_3 p_5 - p_1 p_2 p_4 p_5 - p_1 p_3 p_4 p_5 + p_1 p_2 p_3 p_4 p_5$$



Caz	Înainte de reducere	După reducere	Formula de calcul
1			$P_{12} = 1 - q_1 q_2$ $\Omega = 1$ $K' = K$
2			$P_{12} = p_1 p_2$ $\Omega = 1$ $K' = K$
3		$v \neq s$	$P_{12} = \frac{p_1 p_2}{\Omega} = p_2$ $\Omega = p_1$ $K' = K - v - u + t$
4		$v \equiv s$	$P_{12} = 0$ $\Omega = p_2$ $K' = K - v - w + t$
5			$P_{12} = 0$ $\Omega = 1$ $K' = K$
6			
7		$v \neq s$ —	$R(D_K = 0)$
8		$v \equiv s$	$\Omega = p_1$ $K' = K - v - w + t$
9		$v \neq s$ —	
10		$v \equiv s$ —	$R(D_K = 0)$
11			$\Omega = 1$ $K' = K$
12			

Tabelul 3.6.3.

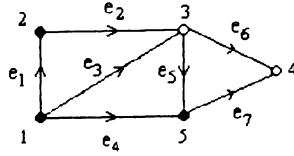


Figura 3.6.4.

**Factorizarea.** Pentru rezolvarea problemei grafurilor serie-paralele ireductibile la calculul fiabilității de tip  $K$ -terminal, precum și a grafurilor care nu sunt serie-paralele, reducerea este posibilă totdeauna prin *factorizare* (factoring). Această tehnică partiționează spațiul de probabilitate în două, în funcție de starea (bună sau defectă) a unei anumite muchii. Dacă  $e_i = (u, v)$  este o muchie a lui  $G_K$ , iar  $A_i$  desemnează evenimentul "e. este funcțional" și  $\bar{A}_i$  evenimentul contrar, atunci conform regulilor de probabilitate condiționată (formula (3.5.12)) se poate scrie:

$$R(G_K) = p_i R(G_K | A_i) + q_i R(G_K | \bar{A}_i) = p_i R(G'_K) + q_i R(G''_K) \quad (3.6.3)$$

În relația de mai sus  $G'_K$  și  $G''_K$  reprezintă grafuri induse din  $G_K$  de către evenimentele  $A_i$ , respectiv  $\bar{A}_i$ . De fapt ambele corespund la eliminarea din graful  $G_K$  a muchiei  $e_i$ : la  $G'_K$  aceasta fiind contractată (nodurile  $u$  și  $v$  contopite), iar la  $G''_K$  fiind ștearsă. Metoda se poate aplica în mod recursiv prin eliminarea a câte unei muchii din graf la fiecare pas (figura 3.6.5). Factorizarea după muchii (edge factoring) se utilizează ca o ultimă posibilitate atunci când nici o altă transformare nu se poate efectua. Dacă se consideră noduri imperfecte, factorizarea se poate aplica și după noduri (vertex factoring) [SHOO92]. Deoarece factorizarea creează pentru fiecare problemă două noi subprobleme, ea contribuie la creșterea exponențială a algoritmului de calcul al fiabilității.

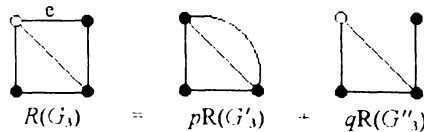


Figura 3.6.5.

Înainte de aplicarea factorizării după o anumită muchie  $e(u, v)$ , trebuie studiat dacă muchia respectivă poate fi luată ca element cheie (keystone) sau nu. Subgraful  $G''$  obținut prin ștergerea lui  $e$  din  $G$  corespunde evenimentului "e defect" în sistemul inițial. În schimb, scurtcircuitarea lui  $e$  poate induce un subgraf  $G'$  care nu întodeauna corespunde evenimentului "e funcțional" din graful  $G$ , dacă muchia  $e$  este orientată. Comasarea nodurilor  $u$  și  $v$  în graful indus creează o cale de comunicație și în sensul în care ea nu a existat în  $G$ , ceea ce introduce o eroare de calcul. Este clar că o muchie neorientată nu creează niciodată asemenea problemă, deci poate fi aleasă ca element cheie. Criteriile pentru muchiile orientate din grafurile orientate sau mixte au fost studiate de Nakazawa, care întrebuițează termenul de *linie eliminabilă* (removable line) pentru o muchie după

care se poate efectua descompunerea lui  $G$  [NAKA81]. El clasifică tipurile de rețele după numărul perechilor de noduri de intrare/ieșire a informației:

- Tipul I - are o singură pereche de noduri de intrare/ieșire  $(s, t)$
- Tipul II - are două sau mai multe perechi de forma  $(s_j, t_j), j = 1, 2, \dots$
- Tipul II' - are două sau mai multe perechi de felul  $(s_j, t), j = 1, 2, \dots$
- Tipul III - are două sau mai multe perechi de tipul  $(s_j, t_j), j = 1, 2, \dots$

În lucrarea menționată se demonstrează că în orice rețea, cu excepția celei orientate de tipul III, există totdeauna linii eliminabile. O muchie orientată  $e(u \rightarrow v)$  este eliminabilă dacă și numai dacă pentru orice pereche de noduri  $(s_j, t)$  se îndeplinește cel puțin una din următoarele condiții:

1.  $u = s_j$
2.  $v = t$
3.  $u, v \neq s_j, t_j$ : nu există nici o cale  $(s_j \rightarrow v)$  care nu trece nici prin  $u$ , nici prin  $t_j$  și nu există nici o cale  $(u \rightarrow t_j)$  care să nu treacă nici prin  $v$ , nici prin  $s_j$ .
4.  $u = t_j, v \neq s_j$ : nu există nici o cale  $(s_j \rightarrow v)$  care nu trece prin  $u$  sau  $t_j$
5.  $u \neq t_j, v = s_j$ : nu există nici o cale  $(u \rightarrow t_j)$  care nu trece prin  $v$  sau  $s_j$

Calculul fiabilității rețelei se face prin descompunere repetată, căutând pe rând linii eliminabile, până când se obțin subgrafuri cu număr minim de muchii, care au fiabilitatea cunoscută sau egală cu zero.

Pentru ca prezentarea să fie completă, studiez în tabelul 3.6.6 modul de descompunere al grafului inițial  $G$  în cele trei cazuri arătate la punctul 3 de mai înainte, când nu se pot face transformări simple. Este de remarcat faptul că nu este necesară factorizarea decât după una din cele două muchii, la alegere, pentru a defalca problema de calcul în două subprobleme tratabile.

Graful inițial	Factorizare după $e_1$	Factorizare după $e_2$
Formula de calcul	$R(G) = p_1 R(G') + q_1 p_2 R(G'')$	$R(G) = p_2 R(G') + p_1 q_2 R(G'')$

Tabelul 3.6.6.

Reducțiile simple și factorizarea sunt suficiente pentru a rezolva problema fiabilității de tip  $K$ -terminal pentru un graf oarecare. În unele situații pentru simplificarea calculului se pot aplica și *transformări triunghi-stea* (delta-star transformations, delta-Y transformations) cu menținerea fiabilității, care reduc gradul nodurilor implicate cu 1 [GROSH89][SHOO92][HWANG81]. O altă reducere, cu aplicabilitate limitată însă numai la grafurile care conțin poligoane, va fi studiată în continuare.

**Reducții poligon-lanț.** Pentru problema grafurilor serie-paralele ireductibile s-a propus de către Satyanarayana și Wood o metodă de reducere bazată pe transformări de tip *lanț din poligon* (poligon-to-chain reductions) [SATY85]. Un lanț constă dintr-o secvență alternantă de noduri și muchii  $v_1, (v_1v_2), v_2, \dots, v_{k-1}, (v_{k-1}v_k), v_k$ , în care toate nodurile interne  $v_2, v_3, \dots, v_{k-1}$  sunt de gradul 2, iar nodurile extreme  $v_1$  și  $v_k$  au grad mai mare ca 2. Un poligon este o buclă formată din două lanțuri care au nodurile extreme comune, toate celelalte noduri având gradul 2. Dacă un graf  $G_K$  conține un poligon, atunci acesta se încadrează, după efectuarea reducărilor simple, într-unul din cele 7 tipuri prezentate în figura 3.6.7. Pentru aceasta este posibilă o transformare într-un aranjament sub formă de lanț cu conservarea fiabilității, formulele de calcul fiind date în lucrarea amintită mai sus. Importanța transformărilor poligon-lanț rezultă din următoarea teoremă [SATY85]:

**Teorema 3.6.3.** Un graf serie-paralel  $G_K$  în care nu se pot efectua reducări simple de tip paralel, serial și de gradul 2, trebuie să admită o reducere de tip poligon-lanț.

*Demonstrație.* Trebuie arătat faptul că  $G$ , un graf serie-paralel, conține un poligon. Fie  $G'$  graful obținut prin substituirea lanțurilor din  $G$  cu câte o singură muchie. Dacă  $G'$  va conține o pereche de muchii paralele, atunci lanțurile corespunzătoare din  $G$  formează un poligon. Graful  $G'$  trebuie să conțină o pereche de muchii paralele, deoarece dacă nu ar conține, atunci nu ar fi posibile reducări simple în  $G'$ , toate nodurile din  $G'$  având grad mai mare ca 2. Aceasta ar însemna că  $G'$  și implicit  $G$ , nu sunt grafuri serie-paralele, ceea ce contrazice ipoteza. □

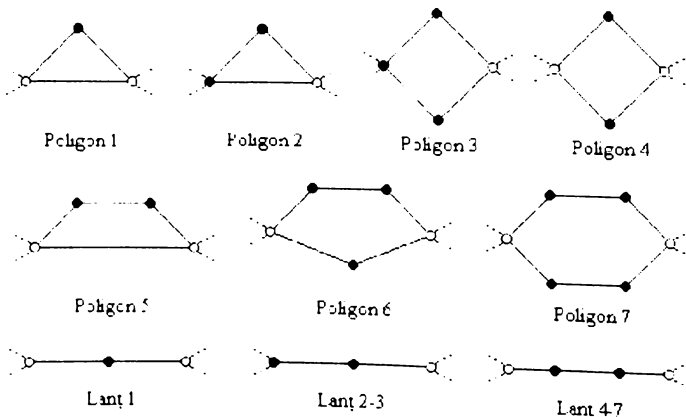


Figura 3.6.7.

Ca urmare a acestei teoreme, orice graf serie-paralel devine reductibil prin aplicarea repetată a reducărilor simple și a celor poligon-lanț, indiferent de nodurile conținute în  $K$ . Mai mult, un astfel de algoritm rezolvă problema de calcul a fiabilității de tip  $K$ -terminal pentru astfel de grafuri într-un timp polinomial  $O(|E|)$ . Acest lucru este posibil deoarece la fiecare pas se induce un singur graf nou, și nu două ca la factorizare. Lucrarea [POLI86] extinde clasa grafurilor pentru care există algoritmi de calcul în timp polinomial și la rețele planare lipsite de cicluri interioare (ICF, inner-cycle free), lipsite de cicluri interioare mai lungi de trei muchii (IFCF, inner-four-cycle-free) și lipsite de cuburi planare (CF, cube-free). Acești algoritmi pot fi utilizați în conjuncție cu tehnica de factorizare pentru calculul fiabilității oricărui graf într-un timp mai scurt decât cel exponențial.

### 3.7. Sisteme de tip multiterminal

În cazul cel mai general, graful probabilistic  $G = (V, E)$  care modelează o rețea de componente (de exemplu calculatoare) nu este de tip serie-paralel și atunci, transformările în graf nefiind posibile, calculul de fiabilitate se bazează pe regulile din teoria probabilităților. Problema tipică de fiabilitate în astfel de sisteme constă în a determina probabilitatea ca o pereche de noduri  $s$ - $t$  (source-target) să poată comunica prin cel puțin o cale de legături și noduri funcționale, respectiv, în caz mai general, probabilitatea ca o anumită submulțime  $K \subseteq V$  de noduri să poată comunica între ele. Primul caz reprezintă o *fiabilitate de tip terminal* (2-terminal reliability, terminal reliability) pe când al doilea intră în categoria *fiabilităților de tip multiterminal* (multiterminal reliability) având ca și caz particular fiabilitatea de ansamblu a rețelei (network reliability, overall reliability, all-terminal reliability) pentru  $K=V$  [RAI90a]. În modelul matematic aferent se admite că legăturile sunt expuse la defectări aleatoare independente, având probabilități de bună funcționare  $p_i$ , respectiv de nefuncționare  $q_i = 1 - p_i$ , cunoscute [VISW87]. Există în principal patru clase de metode pentru calculul fiabilității sistemelor multiterminal, în funcție de formula de probabilitate aleasă pentru calcul: enumerare, incluziune-excluziune, sumă de produse disjuncte și descompunere după un element cheie.

**Enumerarea stărilor.** Aceasta este metoda cea mai naturală, deoarece constă în enumerarea tuturor stărilor posibile ale sistemului (în total  $2^n$ ) și reținerea acelor care satisfac criteriul de funcționalitate asociat indicatorului de fiabilitate ales. Este o metodă de căutare exhaustivă (exhaustive search). Funcția de structură  $\Phi$  se exprimă sub formă booleană ca reuniunea de termeni corespunzători tuturor stărilor favorabile, de aceea metoda se mai numește de *dezvoltare canonică directă* (direct canonical expansion). Fiecare stare fiind distinctă, evenimentele corespunzătoare sunt reciproc exclusive (disjuncte), fiabilitatea sistemului fiind calculabilă ca o sumă de probabilități:

$$R = \Pr(\Phi = 1) = \Pr\left(\bigcup_{i=1}^m A_i\right) = \sum_{i=1}^m \Pr(A_i) \quad (3.7.1)$$

În relația (3.7.1)  $A_i$  reprezintă evenimentul corespunzător unei stări operaționale a sistemului, fiind de forma  $A_i = \bar{x}_1 \cap \bar{x}_2 \cap \dots \cap \bar{x}_n$  (unde  $\bar{x}_c = x_c$  dacă componenta  $c$  este bună, respectiv  $\bar{x}_c = \bar{x}_c$  dacă este defectă), iar  $m$  reprezintă numărul total al acestor stări. Se observă că metoda devine greoaie dacă  $n$  este mare ( $n > 5$ ).

**Exemplul 3.7.1.** Graful în punte (bridge network) din figura 3.7.1 reprezintă o rețea de referință frecvent utilizată în literatura de specialitate pentru ilustrarea algoritmilor de determinare a fiabilității de tip terminal. Variabilele de stare asociate setului de muchii  $\{e_1, e_2, \dots, e_5\}$  sunt  $\{x_1, x_2, \dots, x_5\}$ . Sistemul este operațional ( $s$  comunică cu  $t$ ), în 16 din cele 32 de stări posibile ale sale, funcția de structură corespunzătoare fiind:

$$\begin{aligned} \Phi = & \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 x_5 \cup \bar{x}_1 x_2 \bar{x}_3 x_4 x_5 \cup \bar{x}_1 x_2 x_3 \bar{x}_4 x_5 \cup \bar{x}_1 x_2 x_3 x_4 \bar{x}_5 \cup \\ & \cup \bar{x}_1 x_2 x_3 x_4 x_5 \cup x_1 \bar{x}_2 \bar{x}_3 x_4 \bar{x}_5 \cup x_1 \bar{x}_2 \bar{x}_3 x_4 x_5 \cup x_1 \bar{x}_2 x_3 \bar{x}_4 x_5 \cup \\ & \cup x_1 \bar{x}_2 x_3 x_4 \bar{x}_5 \cup x_1 \bar{x}_2 x_3 x_4 x_5 \cup x_1 x_2 \bar{x}_3 \bar{x}_4 x_5 \cup x_1 x_2 \bar{x}_3 x_4 \bar{x}_5 \cup \\ & \cup x_1 x_2 \bar{x}_3 x_4 x_5 \cup x_1 x_2 x_3 \bar{x}_4 x_5 \cup x_1 x_2 x_3 x_4 \bar{x}_5 \cup x_1 x_2 x_3 x_4 x_5 \end{aligned} \quad (3.7.2)$$

(Pentru a simplifica notația, în loc de  $\bar{x}_1 \cap x_2 \cap \bar{x}_3 \cap \bar{x}_4 \cap x_5$  am scris  $\bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 x_5$ , ș.a.m.d.). Conform relației (3.7.1), funcția de fiabilitate se exprimă ca o sumă de probabilități, în care fiecare termen conține variabile asociate unor evenimente independente, deci  $\Pr(\bar{x}_1 \bar{x}_2 \dots \bar{x}_5) = \Pr(\bar{x}_1) \Pr(\bar{x}_2) \dots \Pr(\bar{x}_5)$ . Folosind notația  $\Pr(x_i) = p_i$  și  $\Pr(\bar{x}_i) = q_i$ , rezultă că fiabilitatea rețelei se obține înlocuind direct în (3.7.2) pe  $x_i$  cu  $p_i$ , pe  $\bar{x}_i$  cu  $q_i$  și operațiile booleene cu operații aritmetice:

$$\begin{aligned} R_{\pi} = & q_1 p_2 q_3 q_4 p_5 + q_1 p_2 q_3 p_4 p_5 + q_1 p_2 p_3 q_4 p_5 + q_1 p_2 p_3 p_4 q_5 + \\ & + q_1 p_2 p_3 p_4 p_5 + p_1 q_2 q_3 p_4 q_5 + p_1 q_2 q_3 p_4 p_5 + p_1 q_2 p_3 q_4 p_5 + \\ & + p_1 q_2 p_3 p_4 q_5 + p_1 q_2 p_3 p_4 p_5 + p_1 p_2 q_3 q_4 p_5 + p_1 p_2 q_3 p_4 q_5 + \\ & + p_1 p_2 q_3 p_4 p_5 + p_1 p_2 p_3 q_4 p_5 + p_1 p_2 p_3 p_4 q_5 + p_1 p_2 p_3 p_4 p_5 \end{aligned} \quad (3.7.3)$$

□

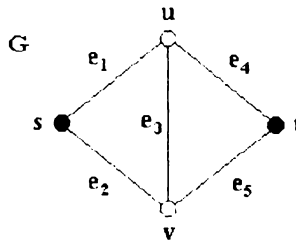


Figura 3.7.1.

**Principiul de incluziune-excluziune.** Pentru a elimina dezavantajul cauzat de enumerarea unui număr mare de stări, calculul de fiabilitate se poate face pornind de la un set minim de arbori de acoperire  $(T_1, T_2, \dots, T_m)$ . Un *arbore de acoperire* (spanning tree) este un subgraf conex al lui  $G$  care conține toate nodurile din  $K$ , deci le interconectează asigurând astfel funcționalitatea sistemului. Arborele de acoperire minimal este acela care nu mai conține un alt arbore de acoperire. În acest caz funcția de structură  $\Phi$  se va scrie ca o sumă booleană a termenilor corespunzători arborilor de acoperire minimali, iar funcția de fiabilitate se calculează cu formula de *incluziune-excluziune* (inclusion-exclusion) din teoria probabilităților:

$$\begin{aligned}
R = \Pr(\Phi = 1) &= \Pr\left(\bigcup_{i=1}^m A_i\right) = \sum_{i=1}^m \Pr(A_i) - \sum_{i < j} \Pr(A_i \cap A_j) + \\
&+ \sum_{i < j < k} \Pr(A_i \cap A_j \cap A_k) - \dots + (-1)^{m-1} \Pr(A_1 \cap A_2 \cap \dots \cap A_m)
\end{aligned} \tag{3.7.4}$$

Aici  $A_i$  reprezintă evenimentul corespunzător arborelui  $T_i$  funcțional, aceste evenimente nefiind reciproc exclusive.

În particular, pentru fiabilitatea de tip terminal între o pereche de noduri  $s-t$ , arborele de acoperire minimal se reduce la o cale între  $s$  și  $t$ . Calculul se face atunci pornind de la un set de căi minimale (minimal path set), de aceea metoda se mai numește prin *enumerare de căi* (path tracing). Dacă există în total  $m$  căi, atunci formula (3.7.4) implică  $2^m - 1$  termeni.

**Exemplul 3.7.2.** Pentru cazul din figura 3.7.1, între nodurile  $s$  și  $t$  există patru căi posibile, funcția de structură a sistemului fiind:

$$\Phi = x_1 x_4 \cup x_2 x_5 \cup x_1 x_3 x_5 \cup x_2 x_3 x_4 \tag{3.7.5}$$

Folosind relația (3.7.4) și având în vedere proprietatea de idempotență a variabilei binare, funcția de fiabilitate corespunzătoare va fi:

$$\begin{aligned}
R_{st} = \Pr(\Phi=1) &= \Pr(x_1 x_4) + \Pr(x_2 x_5) + \Pr(x_1 x_3 x_5) + \Pr(x_2 x_3 x_4) - \\
&- \Pr(x_1 x_2 x_4 x_5) - \Pr(x_1 x_3 x_4 x_5) - \Pr(x_1 x_2 x_3 x_4) - \Pr(x_1 x_2 x_3 x_5) - \\
&- \Pr(x_2 x_3 x_4 x_5) - \Pr(x_1 x_2 x_3 x_4 x_5) + \Pr(x_1 x_2 x_3 x_4 x_5) + \\
&+ \Pr(x_1 x_2 x_3 x_4 x_5) + \Pr(x_1 x_2 x_3 x_4 x_5) + \Pr(x_1 x_2 x_3 x_4 x_5) - \\
&- \Pr(x_1 x_2 x_3 x_4 x_5) = p_1 p_4 + p_2 p_5 + p_1 p_3 p_5 + p_2 p_3 p_4 - \\
&- p_1 p_2 p_4 p_5 - p_1 p_3 p_4 p_5 - p_1 p_2 p_3 p_4 - p_1 p_2 p_3 p_5 - p_2 p_3 p_4 p_5 + \\
&+ 2p_1 p_2 p_3 p_4 p_5
\end{aligned} \tag{3.7.6}$$

Evident expresia obținută este echivalentă cu (3.7.3).

□

Faptul că relația (3.7.4) conține termeni cu semne contrare, a fost exploatat în [SATY81] pentru a obține o formulă topologică simplă pentru fiabilitatea SMT a unei rețele cu conexiuni orientate. Fiabilitatea SMT (source-to-multiple-terminal) desemnează aici probabilitatea ca un anumit nod să fie în legătură cu oricare din celelalte noduri. Studiul se efectuează pe un sistem modelat printr-un  $t$ -graf, care reprezintă un graf conex orientat cu un nod  $s$  numit sursă, având proprietatea că fiecare arc este așezat pe un arbore orientat care pornește din  $s$  (figura 3.7.2). Un arbore orientat (directed tree, rooted tree) este un graf conex lipsit de cicluri având proprietatea că numai un nod, și anume cel sursă, are gradul de incidență spre interior 0, celelalte având gradul 1. Într-un astfel de arbore există numai câte o singură cale de la rădăcină la fiecare alt nod. Mulțimea arborilor de acoperire (care conțin toate nodurile) ai lui  $G$  formează setul minimal de evenimente corespunzătoare succesului. Drept urmare, pentru un graf orientat dat  $G$ , fiabilitatea SMT  $R_s(G)$  referitor la nodul rădăcină  $s$ , constă în probabilitatea ca să existe cel puțin un arbore orientat acoperitor care pornește din  $s$  având toate arcele (și nodurile, dacă se lucrează cu noduri imperfecte) operaționale. Pe baza formulei de incluziune-excluziune,  $R_s(G)$  se poate exprima ca:

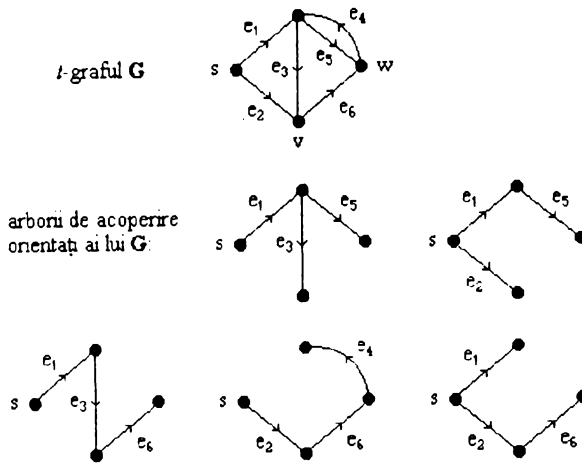


Figura 3.7.2.

$$R_s(G) = \sum_i \varepsilon_i \Pr\{U_i\} \quad (3.7.7)$$

unde  $U_i$  reprezintă uniunea celui de-al  $i$ -lea set de arbori de acoperire a lui  $G$ , iar  $\varepsilon_i$  este +1 sau -1 în funcție de numărul de arbori din  $U_i$  (dacă este un număr par, respectiv impar). Suma se efectuează pentru toate uniunile posibile ale arborilor lui  $G$ , numărul de termeni fiind, așa cum am mai arătat  $2^m - 1$ , unde  $m$  este numărul de arbori de acoperire din  $G$ . Corespunzător tuturor combinațiilor date de arborii de acoperire, pentru  $t$ -graful  $G$  există de fapt  $2^m - 1$   $t$ -subgrafuri (ciclice sau aciclice), dar nu neapărat toate distincte. Dacă se întâmplă ca  $U_i = U_j$  pentru  $i \neq j$  și  $\varepsilon_i = -\varepsilon_j$ , atunci termenii  $\Pr\{U_i\}$  și  $\Pr\{U_j\}$  se anulează. Întrucât fiecare  $U_i$  corespunde unui  $t$ -subgraf al lui  $G$ , formula (3.7.7) se poate rescrie:

$$R_s(G) = \sum_i d_a \Pr\{G_{a,i}\} + \sum_i d_c \Pr\{G_{c,i}\} \quad (3.7.8)$$

unde  $G_{a,i}$  reprezintă al  $i$ -lea  $t$ -subgraf aciclic,  $G_{c,i}$  al  $i$ -lea  $t$ -subgraf ciclic, iar  $d_a$  și  $d_c$  reprezintă dominanta (diferența dintre numărul de formațiuni impare și cele pare) corespunzătoare celor două tipuri de subgrafuri. Se arată că  $d_c = 0$ , deci termenii care rămân în expresie corespund unu la unu cu  $t$ -subgrafurile aciclice (lipsite de cicluri) ale lui  $G$ . Formula de calcul a fiabilității SMT devine:

$$R_s(G) = \sum_i (-1)^{n_i - N_i + 1} \Pr\{G_{a,i}\} \quad (3.7.9)$$

unde  $n_i$  și  $N_i$  reprezintă numărul de arce, respectiv de noduri ale celui de-al  $i$ -lea subgraf aciclic  $G_{a,i}$  al lui  $G$ ,  $\Pr\{G_{a,i}\}$  fiind probabilitatea ca toate arcele (și nodurile) acestuia să fie funcționale. Relația (3.7.9) este general valabilă atât pentru rețele cu legături orientate, cât



și pentru cele neorientate. Pe baza ei se poate calcula fiabilitatea globală a grafului neorientat  $G$ , care este identică în această situație cu fiabilitatea SMT a lui  $G$  alegând oricare nod drept sursă. (Un graf neorientat se poate transforma oricând într-unul orientat, înlocuind muchiile cu câte două arce antiparalele, fiecare având aceeași probabilitate de defectare ca și muchia inițială, dar defectarea uneia implică și defectarea arcului pereche. Se obține un digraf simetric, în care defectările arcelor nu mai sunt  $s$ -independente).

**Suma de produse disjuncte.** Numărul de termeni implicați în formula de calcul a fiabilității se poate reduce dacă se utilizează probabilități condiționate. Funcția de structură are aceeași exprimare ca la metoda precedentă, bazată pe setul de arbori de acoperire minimali (respectiv căi, în cazul fiabilității corespunzătoare conectivității  $s-t$ ). Evenimentele  $A_i$  corespunzătoare celor  $m$  arbori (căi) nu sunt însă reciproc exclusive, și de aceea pentru calculul funcției de fiabilitate este necesară descompunerea lor în evenimente disjuncte. Întrucât din funcția de structură se determină o sumă de termeni disjuncti (disjoint terms), metoda se mai numește de *extragere algebrică* (algebraic extraction) a termenilor exclusivi. Formula de calcul a fiabilității este:

$$\begin{aligned} R = \Pr(\Phi = 1) &= \Pr\left(\bigcup_{i=1}^m A_i\right) = \Pr(A_1) + \Pr(A_2) \cdot \Pr(\bar{A}_1 | A_2) + \dots + \\ &+ \Pr(A_m) \cdot \Pr(\bar{A}_1 \cap \bar{A}_2 \cap \dots \cap \bar{A}_{m-1} | A_m) = \Pr(A_1) + \\ &+ \Pr(A_2 \cap \bar{A}_1) + \dots + \Pr(A_m \cap \bar{A}_1 \cap \bar{A}_2 \cap \dots \cap \bar{A}_{m-1}) \end{aligned} \quad (3.7.10)$$

În această relație  $\Pr(\bar{A}_1 | A_2)$  reprezintă probabilitatea ca  $T_1$  să fie în stare defectă cu condiția ca  $T_2$  să fie în stare funcțională. Se observă că expresia de fiabilitate operează numai cu  $m$  termeni, dar calculul este de regulă foarte dificil (exponențial cu  $m$ ), deoarece termenii trebuie să corespundă la evenimente disjuncte. Problema disjunctizării termenilor fiind importantă sub aspectul contribuțiilor aduse, îi voi dedica un paragraf separat.

**Exemplul 3.7.3.** Revin la figura 3.7.1 pentru a ilustra modalitățile de calcul a fiabilității corespunzătoare nodurilor  $s-t$  pe baza relației (3.7.10) pornind de la setul de căi minimale  $\{x_1x_4, x_2x_5, x_1x_3x_5, x_2x_3x_4\}$ . Funcția de structură are aceeași expresie (3.7.5). Expresia fiabilității sub formă de sume disjuncte va fi:

$$\begin{aligned} R_{s,t} &= \Pr(x_1x_4) + \Pr(x_2x_5) + \Pr(x_1x_3x_5) + \Pr(x_2x_3x_4) + \Pr(x_1x_2x_3x_4) + \Pr(x_1x_2x_3x_4x_5) = \\ &= \Pr(x_1x_4) + \Pr[x_2x_5(\bar{x}_1 \cup \bar{x}_4)] + \Pr[x_1x_3x_5(\bar{x}_1\bar{x}_2 \cup \bar{x}_2\bar{x}_4 \cup \bar{x}_1\bar{x}_5 \cup \bar{x}_4\bar{x}_5)] + \\ &+ \Pr[x_2x_3x_4(\bar{x}_1\bar{x}_2 \cup \bar{x}_1\bar{x}_5 \cup \bar{x}_2\bar{x}_3\bar{x}_4 \cup \bar{x}_4\bar{x}_5)] = \Pr(x_1x_4) + \Pr(\bar{x}_1x_2x_5 \cup x_1x_2\bar{x}_4x_5) + \\ &+ \Pr(x_1\bar{x}_2x_3\bar{x}_4x_5) + \Pr(\bar{x}_1x_2x_3x_4\bar{x}_5) = p_1p_4 + q_1p_2p_5 + p_1p_2q_4p_5 + p_1q_2p_3q_4p_5 + \\ &+ q_1p_2p_3p_4q_5 \end{aligned} \quad (3.7.11)$$

Expresia obținută este evident echivalentă cu (3.7.3) și (3.7.6).  $\square$

Pentru calculul fiabilității de tip terminal trebuie luate în considerare toate combinațiile posibile de legături care formează căi de comunicație între sursă și destinație. Fiabilitatea globală a rețelei se poate calcula folosind acest concept al fiabilității de tip

terminal, prin determinarea tuturor căilor posibile între fiecare din cele  $N(N-1)/2$  perechi de noduri ( $N$  fiind numărul total de noduri). Pentru un graf cu număr mare de noduri această metodă nu este însă eficientă, o soluție alternativă fiind cea bazată pe arborii de acoperire. În acest caz un arbore de acoperire  $T_i$  va conține toate cele  $n$  noduri ale lui  $G$  și pentru a le interconecta se folosesc  $N-1$  muchii, care reprezintă numărul minim de legături necesare pentru realizarea comunicației între toate nodurile. Pentru studiul fiabilității globale sunt necesare următoarele etape:

- a) determinarea tuturor arborilor minimali  $T_i$  ai grafului
- b) interpretarea expresiei booleene obținute la a) sub forma unei relații de probabilitate

O metodă de determinare a arborilor de acoperire minimali se bazează pe generarea ciclică a acestora pornind de la un arbore arbitrar. O altă metodă, implementabilă pe calculator, constă în efectuarea produsului cartezian a  $N-1$  tăieturi de vîrf  $C_i$ , ale căror elemente constau din muchiile conectate la cele  $N-1$  noduri alese arbitrar [AGGA81]. Pentru puntea dată ca exemplu în figura 3.7.1, se obține:

$$C = C_1 \times C_2 \times \dots \times C_{N-1} = (x_1, x_2) \times (x_4, x_5) \times (x_1, x_3, x_4) = (x_1 x_3 x_4, x_1 x_3 x_5, x_1 x_4 x_5, x_1 x_2 x_4, x_2 x_3 x_4, x_1 x_2 x_5, x_2 x_3 x_5, x_2 x_4 x_5) = \{T_i\}_{i=0, \dots, n-1}$$

Pentru a satisface criteriul de fiabilitate stabilit, situația de reușită corespunde evenimentului de a avea cel puțin un arbore cu muchiile funcționale, adică:

$$S = T_0 \cup T_1 \cup \dots \cup T_{n-1}$$

Această expresie booleană nu conține însă termeni disjuncți care să corespundă la evenimente incompatibile astfel încât relația să poată fi transpusă direct în mărime de probabilitate. De exemplu, evenimentele  $T_0$  și  $T_1$  se îndeplinesc simultan dacă sistemul se află în starea în care numai  $e_2$  este defect. Dacă s-a obținut o expresie de termeni disjuncți pentru  $S$ , fiabilitatea rețelei este dată de

$$R_s = S(\text{disjunct})_{x_i \rightarrow p_i, x_j \rightarrow q_j}$$

în care variabilele  $x_i$  și  $x_j$  s-au înlocuit cu probabilitatea de bună funcționare ( $p_i$ ), respectiv de defectare ( $q_j$ ) a arcelor. Considerând toate arcele de aceeași fiabilitate  $p$ , se obține un polinom în  $p$ , care pentru rețeaua punte are expresia [AGGA81]:

$$R_s = 8p^3 - 11p^4 + 4p^5 \quad (3.7.12)$$

Este interesant de remarcat faptul că coeficientul termenului de putere minimă din polinomul  $R_s$  (adică 8) este egal cu numărul arborilor de acoperire minimali ai grafului  $G$ , iar gradul acestui termen (3) reprezintă numărul de muchii ai fiecărui arbore. Dacă polinomul se rescrie ca un polinom în  $(p-1)$ , schimbând pe  $p$  cu  $[(p-1)+1]$  și dezvoltând, atunci se obține expresia

$$R_s = 1 - 2(p-1)^2 + 4(p-1)^3 + 9(p-1)^4 + 4(p-1)^5 \quad (3.7.13)$$

în care termenul de grad minim (excluzând constanta) are proprietatea că exponentul său (2) reprezintă conectivitatea de muchie a lui  $G$ , (adică numărul minim de muchii a căror eliminare deconectează graful), iar valoarea absolută a coeficientului său (2) este egală cu numărul tăieturilor cu cele mai puține muchii [PAGE94]. Reprezentarea grafică a fiabilității  $R_S$  în funcție de  $p$  este dată în figura 3.7.3. Se observă că există o valoare de prag  $p_0$  astfel încât  $R_S(p_0) = p_0$ ,  $R_S(p) \leq p$  pentru  $0 \leq p \leq p_0$  și  $R_S(p) > p$  pentru  $p_0 < p \leq 1$ . Implicația practică a acestui rezultat este că fiabilitatea globală a unei rețele cu structură coerentă devine mai mare decât fiabilitatea unei componente (legături) numai dacă toate acestea au atins o fiabilitate suficient de mare. Dacă se iau în considerare și fiabilitățile nodurilor ( $p_{v_1}, p_{v_2}, \dots, p_{v_N}$ ), atunci fiecare termen  $T_i$  este multiplicat cu același factor și în consecință  $R_S$  se va multiplica cu un factor  $p_{v_1} p_{v_2} \dots p_{v_N}$ , obținându-se:

$$R_S' = p_{v_1} p_{v_2} \dots p_{v_N} R_S \quad (3.7.14)$$

Această relație este însă valabilă numai în cazul fiabilității globale a rețelei.

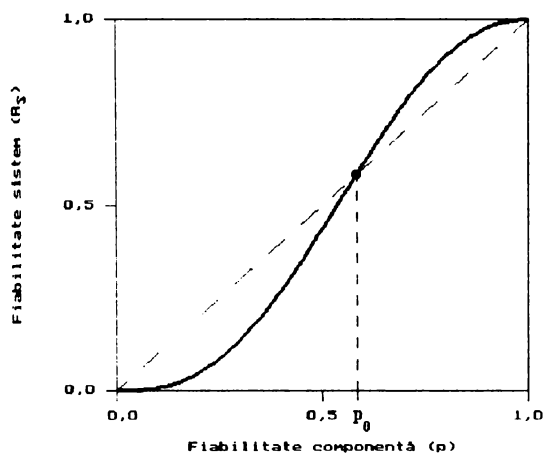


Figura 3.7.3.

**Descompunerea după un element cheie.** Această tehnică, numită și *descompunere pivotală* (pivotal decomposition), constă în scrierea unei relații de probabilitate condiționată în funcție de o anumită componentă “cheie”. De regulă se alege componenta cea mai complicată, adică aceea care împiedică descompunerea sistemului în structuri simple de tip serial și paralel, sau cea care nu permite tratarea independentă a acestor structuri. Funcția de structură  $\Phi$  se descompune astfel în două subfuncții, în funcție de starea elementului pivot  $x_i$ . Relația de calcul a fiabilității este:

$$R = \Pr(\Phi = 1) = p_i \Pr(\Phi | x_i = 1) + (1 - p_i) \Pr(\Phi | x_i = 0) \quad (3.7.15)$$

Se observă că problema de fiabilitate se defalcă în două subprobleme, care pot fi descompuse mai departe alegând pe rând și alte elemente pivot. Calculul este laborios, dar poate fi aplicat la orice sistem cu multe componente.

**Exemplul 3.7.4.** Dacă se alege muchia  $e_3$  ca element pivot pentru rețeaua din figura 3.7.1, fiabilitatea corespunzătoare conectivității  $s-t$  având funcția de structură (3.7.5) se va calcula pe baza (3.7.15) ca:

$$\begin{aligned}
 R_{s,t} &= p_3 \Pr(x_1 x_4 \cup x_2 x_5 \cup x_1 x_5 \cup x_2 x_4) + (1 - p_3) \Pr(x_1 x_4 \cup x_2 x_5) = \\
 &= p_3 \Pr[(x_1 \cup x_2)(x_4 \cup x_5)] + (1 - p_3) \Pr(x_1 x_4 \cup x_2 x_5) = \\
 &= p_3 \Pr(x_1 \cup x_2) \Pr(x_4 \cup x_5) + (1 - p_3) \Pr(x_1 x_4 \cup x_2 x_5) = \\
 &= p_3(p_1 + p_2 - p_1 p_2)(p_4 + p_5 - p_4 p_5) + (1 - p_3)(p_1 p_4 + p_2 p_5 - \\
 &\quad - p_1 p_2 p_4 p_5) = p_1 p_4 + p_2 p_5 + p_2 p_3 p_4 + p_1 p_3 p_5 - p_1 p_2 p_3 p_4 - \\
 &\quad - p_1 p_2 p_3 p_5 - p_1 p_3 p_4 p_5 - p_2 p_3 p_4 p_5 - p_1 p_2 p_4 p_5 + 2 p_1 p_2 p_3 p_4 p_5
 \end{aligned} \tag{3.7.16}$$

Relația obținută este identică cu (3.7.6) de la metoda de incluziune-excluziune.  $\square$

Dacă se face descompunerea pe rând după toate componentele, atunci se obține un număr mare de termeni ( $2^n$ ), de aceea operația se oprește când s-au obținut deja structuri suficient de simple pentru a fi tratate cu o altă metodă. Este recomandabilă adoptarea unei strategii conservative (conservative policy) de descompunere decât a uneia exhaustive, în scopul minimizării numărului de termeni disjuncți și a reducerii numărului de muchii paralele [RAI86a].

**Complexitatea calculului.** Metodele prezentate pentru determinarea fiabilității unui sistem de tip multiterminal necesită un efort de calcul care în general este diferit pentru un anumit sistem în funcție de metoda aleasă. Complexitatea calculului este însă influențată și de caracteristicile topologice ale sistemului. O comparație a timpilor de calcul implicați de evaluarea numerică a fiabilității după diferite metode este făcută în [AGGA75]. Evaluarea timpului de calcul se rezumă însă numai la operațiile de adunare și înmulțire cerute de înlocuirea valorilor numerice în expresia simbolică de fiabilitate obținută după metodele care se compară. Consider însă că o astfel de comparație nu poate fi clocventă dacă nu se ține cont și de efortul necesar pentru deducerea expresiei de fiabilitate pe baza structurii. De aceea, am analizat complexitatea calculului sub aspectul conjugat al determinării expresiei de fiabilitate și al evaluării acesteia pentru două rețele alese de referință (benchmarks): rețeaua punte  $N_4^4$  (bridge network) din figura 3.7.1 și rețeaua ARPA modificată  $N_{13}^9$  (modified ARPANET) din figura 3.6.1.b ( $N_i^j$  reprezentând notația convențională pentru rețeaua benchmark cu  $i$  căi și  $j$  tăieturi minimale). Indicatorul urmărit a fost fiabilitatea de tip terminal corespunzătoare perechii de noduri extreme, rezultatele fiind centralizate în tabelul 3.7.4. În continuare voi face câteva considerații privind modul de întocmire al tabelului și apoi voi trage concluziile care se impun.

Metoda	Rețea punte ( $n=5$ )							Rețea ARPA modificată ( $n=9$ )						
	B	C	M	A	T/T <sub>A</sub>	T/T <sub>A</sub>	$\Delta_R, \Delta_e$	B	C	M	A	T/T <sub>A</sub>	T/T <sub>A</sub>	$\Delta_2, \Delta_e$
	(Aggarwal)(propus)							(Aggarwal)(propus)						
Enumerare	-	92	64	15	655	1115	19	-	4613	1632	204	16524	58041	18
Incluziune-excluziune	1	95	26	9	269	829	40	1505	5965	486	89	4949	464147	716
Sumă disjunctă		6	14	4	144	204	4		132	150	24	1524	2712	6

Tabelul 3.7.4.

A	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
1	1	-	-	1	-
2	-	1	-	-	1
3	1	-	1	-	1
4	-	1	1	1	-

a) pentru punte ( $m=4$ )

A	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$
1	1	-	-	1	-	-	-	1	-
2	-	1	-	-	1	-	-	1	-
3	-	1	-	-	-	1	-	-	1
4	1	-	-	1	-	-	-	1	-
5	1	-	1	-	1	-	-	1	-
6	1	-	1	-	-	1	-	-	1
7	-	1	1	1	-	-	-	1	-
8	-	1	-	-	1	-	1	-	1
9	-	1	-	-	-	1	1	1	-
10	1	-	-	1	1	1	-	-	1
11	1	-	1	-	1	-	1	-	1
12	1	-	1	-	-	1	1	1	-
13	-	1	1	1	-	-	1	-	1

b) pentru ARPA modificat ( $m=13$ )

Tabelul 3.7.5.

Am considerat ca punct de pornire că setul de căi minimale între cele două noduri (cu alte cuvinte funcția de structură booleană sub formă minimă disjunctivă) este dată sub forma unui tabel de  $m$  vectori, ca în tabelul 3.7.5.a, respectiv b. La metoda enumerării, cei  $2^m$  vectori corespunzători stărilor posibile ale sistemului trebuie comparați cu vectorii de cale pentru a decide starea de succes sau nu, ceea ce ar necesita teoretic  $m2^m$  comparații. Practic numărul operațiilor de comparare (C) este mai mic, deoarece pentru o anumită stare în momentul în care s-a ajuns la coincidență nu se mai continuă compararea în tabelul vectorilor de cale, iar starea 00...0 nu trebuie generată, fiind fără sens. La evaluarea cu formula de incluziune-excluziune, fiecare termen de forma  $(A_1 \cap A_2 \cap \dots \cap A_k)$  necesită  $k-1$  operații booleene de tip ȘI. Conform relației (3.7.4) dezvoltarea implică  $\binom{m}{k}$  termeni cu  $k$  evenimente, iar  $k$  variază de la 1 la  $m$ , deci numărul

total de operații booleene (B) este  $\sum_{k=1}^m (k-1) \binom{m}{k}$ . Așa cum am mai arătat, în urma

dezvoltării expresia conține de regulă termeni asemenea, a căror reducere necesită operații suplimentare. Pentru aceasta fiecare termen (vector) din cei  $t = 2^m - 1$  obținuți pe baza relației (3.7.4) trebuie comparat pe rând cu cei următori, numărul de comparații fiind  $(t-1) + (t-2) + \dots + 2 + 1 = t(t-1)/2$ . Practic însă numărul de operații de comparare este mai mic, deoarece comparația nu mai trebuie efectuată cu termenii din tabel care au fost deja reduși. La determinarea fiabilității prin sumă de produse disjuncte am aplicat algoritmul clasic al lui Abraham [ABRA79]. Aici fiecare termen făcut disjunct cu un altul generează o mulțime de noi termeni disjuncți, care trebuie să fie comparați cu toți cei precedenți. De pe urma comparării pot rezulta noi termeni disjuncți, dar cei deja acoperiți se anulează. Algoritmul aplicat pentru rețeaua ARPA modificată este prezentat în extenso în Anexa 1. Am avut în vedere o ordonare după cardinalitate a vectorilor de cale, care eficientizează efortul de calcul [SOH93]. Nu am ținut cont de operațiile de contorizare, care apar la implementarea fiecărei metode. Numărul operațiilor aditive (A) și multiplicative (M) se referă la înlocuirea variabilelor din expresia finală a fiabilității obținută după cele trei metode (relațiile (3.7.3), (3.7.6) și (3.7.11)) cu valori numerice. Se remarcă faptul că nu am inclus în tabel și metoda descompunerii pivotale, deoarece nu

există un algoritm unitar de calcul pe baza ei, fiind folosită mai degrabă pentru defalcarea problemei în două mai simple. Dacă totuși se face descompunerea alegând pe rând toate cele  $n$  componente ca element cheie, atunci s-ar obține o enumerare a tuturor stărilor, ceea ce ar încadra în această categorie metoda amintită.

În ceea ce privește timpul de calcul, se poate accepta ca în [AGGA75] (unde în timpul de calcul se ia în considerare numai înmulțirea și adunarea) că pentru un procesor uzual înmulțirea consumă de 10 ori mai mult timp ca adunarea,  $T_M = 10T_A$ . O operație booleană sau de comparare se poate echivala cu o adunare (scădere), dar fiind efectuată asupra unui vector cu  $n$  componente am admis  $T_E = nT_A$ , respectiv  $T_C = nT_A$ . Pe această bază am obținut în tabelul 3.7.4 timpul total de calcul  $T = BT_B + CT_C + MT_M + AT_A$  raportat la cuanta de timp de adunare pentru metodele analizate. Înainte de concluzii, câteva referiri și la precizia de calcul oferită de fiecare metodă.

Dacă se consideră, pentru simplitate, că toate componentele au aceeași fiabilitate ( $p$ ), respectiv nefiabilitate ( $q$ ), expresia fiabilității prin enumerarea stărilor (3.7.3) pentru rețeaua punte devine:

$$R_{\pi} = p^5 + 5p^4q + 8p^3q^2 + 2p^2q^3 \quad (3.7.17)$$

Eroarea absolută  $\Delta_R$  introdusă de expresia lui  $R_{\pi}$  poate fi evaluată pe baza următoarei observații: în cazul sumei erorile absolute ale termenilor se adună, iar în cazul produsului eroarea absolută se obține prin înmulțirea produsului cu suma erorilor relative ale termenilor. Notez cu  $\Delta_e$  eroarea absolută aferentă fiabilității unei componente, adică unei legături a rețelei. Pentru (3.7.17) eroarea absolută este [AGGA75]:

$$\Delta_R = \Delta_e[(5/p)p^5 + 5(4/p + 1/q)p^4q + 8(3/p + 2/q)p^3q^2 + 2(2/p + 3/q)p^2q^3] = \Delta_e(10p^4 + 36p^3q + 30p^2q^2 + 4pq^3) \quad (3.7.18)$$

care pentru fiabilități înalte ( $p \approx 1$ ) poate fi aproximat cu:

$$\Delta_R \approx 10\Delta_e \quad (3.7.19)$$

Pentru expresia (3.7.6) dată de principiul de incluziune-excluziune se obține:

$$\Delta_{\bar{E}} = \Delta_e(10p^3 + 20p^2 + 6p^2 + 4p) \approx 40\Delta_e \quad (3.7.20)$$

iar pentru (3.7.11) dată de suma termenilor disjuncți:

$$\Delta_{\bar{R}} = \Delta_e(p^3 + p^2 + 2p + 2pq + 3p^2q) + 6p^2q^2 + 4p^3q \approx 4\Delta_e \quad (3.7.21)$$

Pe baza unor raționamente similare am calculat raportul  $\Delta_R / \Delta_e$  pentru fiabilitățile date de cele trei metode în cazul rețelei ARPA modificate, rezultatele fiind trecute în tabelul 3.7.4.

Analizând datele furnizate de tabelul comparativ 3.7.4, se desprind următoarele concluzii:

1. Determinarea fiabilității prin enumerarea stărilor sau pe baza formulei de incluziune-excluziune necesită un efort mare de calcul, deoarece așa cum am arătat, prima

operează cu  $2^n - 1$  stări, iar a doua cu  $2^n - 1$  termeni. Este interesant de remarcat faptul că timpul de calcul prin incluziune-excluziune devine mai mare decât cel prin enumerare la rețeaua ARPA, față de situația inversă de la cealaltă rețea de referință. Aceasta se întâmplă datorită faptului că la ARPA numărul căilor crește considerabil, deci  $2^n > 2^n$ . În consecință, ambele metode sunt simple de implementat, dar implică multe calcule, fiind de aceea recomandabile numai în sisteme mici ( $n$  mic). Metoda bazată pe incluziune-excluziune este de preferat numai dacă numărul legăturilor de interconectare este mic, ceea ce duce la un număr redus de căi ( $m$  mic).

2. Eroarea introdusă la calculul fiabilității prin formula de incluziune-excluziune a probabilităților este mult mai mare decât la celelalte metode, deoarece expresia conține numai termeni cu variabile directe  $p$ , care introduc erori mai mari ( $s$ -a presupus  $p \gg q$ ). În orice caz, calculul prin suma de produse disjuncte, când și numărul de termeni este cel mai mic, se dovedește a fi cel mai exact.

3. Determinarea fiabilității pe baza sumei de termeni disjuncți prezintă față de celelalte metode și avantajul unui timp de calcul mai scăzut, ceea ce face ca procedeul să fie general aplicabil la toate sistemele. Cheia succesului, pentru a obține o expresie simbolică cât mai simplă, constă în găsirea și implementarea unui algoritm cât mai eficient de disjunctizare a termenilor.

Determinarea fiabilității sistemelor coerente modelate printr-un graf probabilistic oarecare constituie o problemă dificilă, de categoria celor *NP-hard*. Aceasta înseamnă că este improbabil ca să existe algoritmi polinomiali de rezolvare. Totuși pentru anumite clase particulare de structuri (de exemplu de tip serial-paralel, arbore, sau anumite clase de rețele planare) pot exista algoritmi eficienți polinomiali de calcul ai unor indicatori de fiabilitate [POLI86]. Înainte de a analiza complexitatea calculului de fiabilitate sub aspectul încadrării într-o problemă de tip *NP-hard*, fac o scurtă trecere în revistă a noțiunilor legate de acest concept, pe baza [HWANG93] și [BALL86].

O noțiune cheie folosită pentru a compara eficiența algoritmilor constă în determinarea creșterii timpului de calcul în funcție de dimensiunea problemei. Un algoritm este de tip *polinomial* dacă există un polinom  $P(d)$  astfel încât complexitatea timpului este  $O(P(d))$  oncare ar fi dimensiunea  $d$  a problemei. Asta înseamnă că timpul de calcul  $H(d)$  este limitat asimptotic de un polinom, adică există constantele pozitive  $k$  și  $d_0$  astfel încât  $H(d) < kP(d)$ , pentru toate valorile  $d > d_0$ . Problemele care au algoritmi de rezolvare deterministici de tip polinomial formează *clasa P* (polynomial class). Algoritmii deterministici este acela care poate fi rulat pe un calculator real, deci fiecare pas este univoc definit prin programul rulat. Mulțimea problemelor rezolvabile în timp polinomial cu algoritmi nedeterministici se numește *clasa NP* (nondeterministic polynomial class). Deoarece algoritmi deterministici constituie cazuri speciale ale celor nedeterministici, este clar că  $P \subseteq NP$  (figura 3.7.6), dar rămâne deschisă întrebarea dacă  $P = NP$ ? Problemele din *clasa NP-P* nu sunt tratabile cu calculatorul, decât prin simulare cu algoritmi deterministici, care necesită timp exponențial. Cu alte cuvinte, problemele rezolvabile numai cu algoritmi exponențiali aparțin de *clasa NP*. Există o subclasă de probleme numite *NP-complete* (NPC), care grupează problemele cele mai dificile, având proprietatea că dacă ar exista un algoritm polinomial pentru una din ele, ar implica  $P = NP$ . Pentru versiunea de conținere a unei probleme din *NP* (de exemplu "câte subgrafuri de lui  $G$  există care satisfac proprietatea  $X$ "), față de "câte  $G$  vreun subgraf care satisface proprietatea  $X$ "), s-a definit *clasa #P*. Similar se constituie *clasa problemelor #P-complete*, cu proprietatea, că dacă pentru una din ele ar exista un algoritm polinomial, atunci ar exista pentru toate. Problemele #P-complete sunt cel puțin la fel de dificile decât cele NP-complete, dar invers poate exista o problemă #P-completă, al cărei corespondent să fie în *P*. Oricum problemă care este cel puțin așa de dificilă ca și o problemă NPC este de tip *NP-hard*. Pentru a demonstra că o problemă este *NP-hard* trebuie găsită o reducere în timp polinomial a unei probleme cunoscută a fi *NP-hard* la problema în cauză.

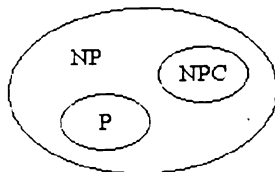


Figura 3.7.6.

O abordare algoritmică a analizei de fiabilitate este cea care pornește cu enumerarea tuturor căilor minimale, respectiv a tăieturilor minimale. Dacă numărul de subgrafuri care satisfac această proprietate (de a fi cale, respectiv tăietură minimală) este foarte mare, atunci și problema de enumerare crește ca și complexitate. În particular, dacă numărul de subgrafuri care satisfac o anumită proprietate, este exponențial cu dimensiunea rețelei, atunci algoritmul de enumerare necesită un timp de calcul exponențial și nu polinomial [BALL86].

Provan și Ball au demonstrat că nu poate exista un algoritm cu limită de timp polinomială în funcție de dimensiunea rețelei (numărul de noduri  $|V|$  și  $|E|$ ) și numărul de căi  $(s, t)$ , numai dacă ar fi adevărat  $P = NP$  [PROV84]. Pentru determinarea setului de căi minimale există algoritmi polinomiali în funcție de  $|V|$  și  $|E|$ , dar problema fiabilității  $(s, t)$ , corespunzătoare este NP-hard. Problema enumerării tăieturilor  $(s, t)$  minimale este #P-completă, în schimb calculul fiabilității corespunzătoare se poate face în timp polinomial față de cardinalitatea mulțimii tăieturilor. În lucrarea menționată anterior se prezintă un astfel de algoritm de complexitate  $O(|V| + |E|\mu^2)$ , unde  $\mu$  reprezintă numărul de tăieturi  $(s, t)$ . Pentru conectivitatea de tip  $K$ -terminal, problemele de fiabilitate aferente sunt NP-hard. În cazul fiabilității globale ( $K=V$ ), grafurile orientate și neorientate trebuie analizate separat. Căile care interconectează nodurile sunt arbori de acoperire orientați cu rădăcina în  $s$ . Fiecare arbore are cardinalitatea  $|V|-1$ , problema determinării acestora fiind rezolvabilă în timp polinomial. Problema tăieturilor minimale poate fi redusă la problema numărării tăieturilor minimale orientate referitoare la  $s$ , care este de tip #P-completă. Enumerarea căilor dintr-un graf neorientat se poate face similar ca la cel orientat în timp polinomial. Numărul de tăieturi minimale este limitat la  $\binom{|V|}{2}$  care pot fi de asemenea enumerate în timp polinomial. Calculul fiabilității globale a rețelei reprezintă însă o problemă NP-hard, în funcție de dimensiunea acesteia atât la grafurile orientate, cât și la cele neorientate, dar în ambele cazuri este posibilă evaluarea în funcție de numărul de căi minimale printr-un algoritm polinomial.

### 3.8. Sisteme duale; căi și tăieturi

Fiind dat un sistem  $S$  bivalent cu  $n$  componente, care are funcția de structură  $\Phi(x_1, x_2, \dots, x_n)$ , se poate defini un sistem dual asociat în felul următor:

**Definiția 3.8.1.** Se numește *sistem dual* (dual system) al unui  $S$ , sistemul notat  $S^d$  care are aceeași mulțime de componente și în care funcția de structură  $\Phi^d: \{0,1\}^n \rightarrow \{0,1\}$  este definită prin:

$$\Phi^d(x_1, x_2, \dots, x_n) = 1 - \Phi(1 - x_1, 1 - x_2, \dots, 1 - x_n) \quad (3.8.1)$$



Întrucât funcția de structură poate fi exprimată și sub formă booleană, relația (3.8.1) se poate scrie ca:

$$\Phi^d(x_1, x_2, \dots, x_n) = \overline{\Phi(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)} \quad (3.8.2)$$

Între sistemul  $S$  și cel dual  $S^d$  există următoarele proprietăți care rezultă pe baza definiției și sunt importante din punct de vedere al fiabilității [VISW87]:

1.  $(S^d)^d = S$ .
2. O cale în  $S$  constituie o tăietură în  $S^d$  și invers.
3. O tăietură în  $S$  constituie o cale în  $S^d$  și invers.
4. O cale minimală în  $S$  constituie o tăietură minimală în  $S^d$  și invers.
5. O tăietură minimală în  $S$  constituie o cale minimală în  $S^d$  și invers.

Este evident faptul că dualul unui sistem de tip serial este un sistem paralel și invers. Căile și tăieturile reprezintă submulțimi ale mulțimii componentelor, înzestrate cu calități distincte pentru descrierea funcționării sistemului. Fie sistemul  $S$  cu componentele  $\Gamma = (c_1, c_2, \dots, c_n)$  și funcția de structură  $\Phi(x_1, x_2, \dots, x_n)$ :

**Definiția 3.8.2.** O *cale* (path, pathset, tie-set) reprezintă o submulțime de componente  $P \subseteq \Gamma$ , cu proprietatea că sistemul funcționează dacă toate aceste componente sunt bune și restul componentelor sunt defecte, adică:

$$\Phi(x_1, x_2, \dots, x_n) = 1 \quad \text{unde } x_i = \begin{cases} 1 & \text{pentru } c_i \in P \\ 0 & \text{pentru } c_i \notin P \end{cases}$$

**Definiția 3.8.3.** O *tăietură* (cut, cutset), reprezintă o submulțime de componente  $C \subseteq \Gamma$ , cu proprietatea că sistemul nu funcționează dacă toate aceste componente sunt defecte și restul componentelor sunt bune, adică:

$$\Phi(x_1, x_2, \dots, x_n) = 0 \quad \text{unde } x_i = \begin{cases} 0 & \text{pentru } c_i \in C \\ 1 & \text{pentru } c_i \notin C \end{cases}$$

Mulțimea căilor, respectiv tăieturilor unui sistem dat este o mulțime ordonată față de relația de incluziune. Fiecărui  $n$ -uplu de stări al componentelor  $X = (x_1, x_2, \dots, x_n)$  îi corespunde ori o cale, ori o tăietură. O cale  $P$  pentru care nu există o submulțime  $P' \subset P$  care să fie de asemenea o cale, reprezintă o *cale minimală* (minimal path, minpath). În mod similar, dacă pentru o tăietură  $C$  nu există o submulțime  $C' \subset C$  care să fie de asemenea o tăietură, atunci aceasta constituie o *tăietură minimală* (minimal cut, mincut). Sistemele coerente pot fi descrise complet fie prin colecția căilor minimale, fie prin colecția tăieturilor minimale. Rezultă că funcția de structură  $\Phi$  a sistemului  $S$  se poate exprima sub forma:

$$\Phi(X) = \bigcup_{P \in \mathcal{P}} \bigcap_{c_i \in P} x_i \quad (3.8.3)$$

unde  $\mathcal{P}$  reprezintă mulțimea căilor minimale, iar  $x_i$  este variabila asociată componentei  $c_i$  din calea minimală  $P$ . Întrucât sistemul nu funcționează ( $\bar{\Phi}=1$ ) dacă există cel puțin o tăietură cu toate componentele sale defecte ( $\bar{x}_i=1, c_i \in C$ ), complementul funcției de structură se poate exprima pe baza tăieturilor minimale care formează mulțimea  $\mathcal{C} = \{C\}$  sub forma:

$$\bar{\Phi}(X) = \bigcup_{C \in \mathcal{C}} \bigcap_{c_i \in C} \bar{x}_i \quad (3.8.4)$$

Funcția de structură nenegată va avea în funcție de variabilele asociate tăieturilor expresia:

$$\Phi(X) = \overline{\bigcup_{C \in \mathcal{C}} \bigcap_{c_i \in C} \bar{x}_i} = \bigcap_{C \in \mathcal{C}} \overline{\bigcap_{c_i \in C} \bar{x}_i} = \bigcap_{C \in \mathcal{C}} \bigcup_{c_i \in C} x_i \quad (3.8.5)$$

Calculul fiabilității sistemului se poate face atât pe baza setului de căi minimale, cât și pe baza setului de tăieturi minimale. Așa cum s-a văzut în paragraful precedent, pentru dezvoltarea relației de probabilitate este mai comod de utilizat o formă disjunctivă a funcției de structură. De aceea, formulele de calcul ale funcției de fiabilitate vor fi:

- pe baza căilor minimale

$$R = \Pr\{\Phi(X) = 1\} = \Pr\left\{\bigcup_{P \in \mathcal{P}} \bigcap_{c_i \in P} x_i\right\} \quad (3.8.6)$$

- pe baza tăieturilor minimale

$$R = \Pr\{\Phi(X) = 1\} = 1 - \Pr\{\bar{\Phi}(X) = 1\} = 1 - \Pr\left\{\bigcup_{C \in \mathcal{C}} \bigcap_{c_i \in C} \bar{x}_i\right\} \quad (3.8.7)$$

Pentru a respecta rigurozitatea prezentării, menționez că în relațiile anterioare  $x_i$  se confundă cu evenimentul "componenta  $c_i$  este bună", iar  $\bar{x}_i$  cu evenimentul "componenta  $c_i$  este defectă". Sub formă algebrică, funcția de structură se poate exprima în funcție de căile minimale, respectiv tăieturile minimale, pe baza definițiilor acestora, ca:

$$\Phi(X) = 1 - \prod_{P \in \mathcal{P}} \left(1 - \prod_{c_i \in P} x_i\right) \quad (3.8.8)$$

$$\Phi(X) = \prod_{C \in \mathcal{C}} \left[1 - \prod_{c_i \in C} (1 - x_i)\right] \quad (3.8.9)$$

Întrucât conform (3.8.2)  $\bar{\Phi}(X) = \Phi^d(\bar{X})$ , iar  $\Pr(x_i)=p_i$ ,  $\Pr(\bar{x}_i)=q_i$ , fiabilitatea sistemului  $S$  se poate determina dacă se cunoaște expresia fiabilității sistemului dual, în care se interschimbă fiabilitatea componentelor cu nefiabilitățile lor:

$$R = \Pr\{\Phi(X) = 1\} = 1 - \Pr\{\Phi^d(\bar{X}) = 1\} = 1 - R_{p_i \rightarrow q_i, q_i \rightarrow p_i}^d \quad (3.8.10)$$

**Exemplul 3.8.4.** În figura 3.8.1, respectiv tabelul 3.8.2, se dă graful dual al rețelei punte considerată deja în figura 3.7.1, împreună cu mulțimile de căi și tăieturi minimale din cele două grafuri. (Se remarcă faptul că există mulțimi de componente care constituie atât cale minimală, cât și tăietură minimală). Pornind de la căile minimale din sistemul dual, fiabilitatea aferentă comunicației între perechea de noduri  $(s, t)$  a acestuia este:

$$\begin{aligned} R_{s,t}^d &= \Pr\{\Phi^d = 1\} = P\{x_1x_2 \cup x_4x_5 \cup x_1x_3x_5 \cup x_2x_3x_4\} = \\ &= p_1p_2 + p_4p_5 + p_1p_3p_5 + p_2p_3p_4 - p_1p_2p_4p_5 - p_1p_2p_3p_5 - \\ &\quad - p_1p_2p_3p_4 - p_1p_3p_4p_5 - p_2p_3p_4p_5 + 2p_1p_2p_3p_4p_5 \end{aligned} \quad (3.8.11)$$

Înlocuind acum în  $R^d$  pe  $p_i$  cu  $q_i=1-p_i$ , pe baza relației (3.8.10) se obține fiabilitatea sistemului original:

$$\begin{aligned} R_{s,t} &= 1 - q_1q_2 - q_4q_5 - q_1q_3q_5 - q_2q_3q_4 + q_1q_2q_4q_5 + q_1q_2q_3q_5 + \\ &\quad + q_1q_2q_3q_4 + q_1q_3q_4q_5 + q_2q_3q_4q_5 - 2q_1q_2q_3q_4q_5 = p_2p_5 + \\ &\quad + p_1p_4 + p_2p_3p_4 + p_1p_3p_5 - p_1p_2p_3p_4 - p_1p_2p_3p_5 - \\ &\quad - p_1p_2p_4p_5 - p_1p_3p_4p_5 - p_2p_3p_4p_5 + 2p_1p_2p_3p_4p_5 \end{aligned} \quad (3.8.12)$$

Se observă că relația este identică cu (3.7.6), obținută prin metoda de incluziune-excluziune direct din rețeaua originală. □

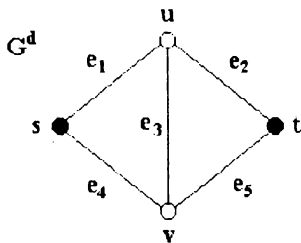


Figura 3.8.1.

G	
Căi minimale	Tăieturi minimale
$e_1, e_4$	$e_1, e_2$
$e_2, e_5$	$e_4, e_5$
$e_1, e_3, e_4$	$e_1, e_3, e_5$
$e_2, e_3, e_4$	$e_2, e_3, e_4$
Tăieturi minimale	Căi minimeale
G <sup>d</sup>	

Tabelul 3.8.2.

În cazul problemei conectivității  $(s, t)$ , căile minimale sunt de fapt drumurile simple de la  $s$  la  $t$ , iar tăieturile minimale sunt mulțimile minimale de elemente a căror eliminare deconectează pe  $t$  de  $s$ . Există multe metode de determinare a căilor dintre două noduri, cum ar fi tehnica de backtracking [PĂTR94], procedura de căutare în lățime (breadth-first search) [BENN82], eliminarea secvențială a nodurilor intermediare până când matricea de conexiune  $M$  se reduce la dimensiunea  $2 \times 2$  [MAYE72], metoda compunerii latine care calculează matricea  $[M]^{i-1}$  [TĂRC89], algoritmul de căutare a căilor minimale bazat pe o descriere a rețelei cu vectori binari [NEVA88] și altele. Uneori trebuie luată o decizie dacă este mai eficient să se facă evaluarea fiabilității pe baza enumerării căilor sau

tăieturilor și acest lucru depinde de structura rețelei. Într-un graf  $G$  oarecare cu  $|E|$  muchii și  $|V|$  noduri, numărul tăieturilor dintre oricare pereche de noduri este de ordinul  $2^{|V|-2}$ , iar numărul căilor de  $2^{|\mathcal{E}|-|V|+2}$ , ceea ce înseamnă o creștere exponențială a acestora odată cu dimensiunea rețelei [WILK72]. Pentru un graf la care nodurile au un grad mediu mai mare de 4, atunci  $|E| > 2|V|$  și  $2^{|\mathcal{E}|-|V|+2} > 2^{|V|-2}$ . Rezultă că un asemenea graf are un număr mai mare de căi decât de tăieturi și deci determinarea nefiabilității, adică  $F = \Pr(\Phi(X) = 1)$ , pe baza tăieturilor este mai simplă în privința timpului de calcul. Relația principală de calcul este (3.8.7). Un algoritm de determinare a fiabilității

$$R_{\pi} = \Pr\{\text{există o cale operațională de la } s \text{ la } t\} = \Pr\{EP(s, t)\}$$

în timp polinomial pornind de la tăieturile  $(s, t)$  ale unui graf  $G$  care poate fi orientat sau neorientat, este următorul [PROV84]:

1. Se enumeră tăieturile minimale  $(s, t)$  ale lui  $G$ . Fiecare tăietură  $C$  partiționează nodurile în două submulțimi disjuncte  $SN(C)$  și  $TN(C)$ . În  $SN(C)$  se definește setul de noduri extreme  $SE(C)$ , de la care există arcuri la  $TN(C)$ .
2. Pentru fiecare tăietură  $C$ , în ordinea crescătoare a cardinalității lui  $SN(C)$ , se evaluează:

$$\Pr\{EC(C)\} = \Pr\{\text{există o cale de la } s \text{ la toate nodurile extreme din } SE(C), \text{ dar la nici un nod din } TN(C)\}$$

Evaluarea se face pe baza lui  $EC(C')$ , unde  $C'$  reprezintă tăieturile  $(s, t')$ ,  $t'$  fiind un nou nod care înlocuiește nodurile  $TN(C)$  și transformă graful într-un graf contractat ( $p_e$  reprezintă fiabilitatea muchiei  $e$ ):

$$\Pr\{EC(C)\} = \prod_{e \in C} p_e \left\{ 1 - \sum_{C' \in \mathcal{E}} \Pr\{EC(C')\} / \prod_{e \in C' \cap C} p_e \right\} \quad (3.8.13)$$

3. Se evaluează

$$\Pr\{EP(s, t)\} = 1 - \Pr\{\overline{EP(s, t)}\} = 1 - \sum_{C \in \mathcal{E}} \Pr\{EC(C)\} \quad (3.8.14)$$

însurarea fiind posibilă deoarece  $\overline{EP(s, t)} = \bigcup EC(C)$  și elementele reuniunii sunt disjuncte. (Disjuncția se demonstrează prin faptul că dacă se produce  $\overline{EP(s, t)}$ , aceasta poate fi cauzată de un singur termen  $EC(C)$ ).

Utilizarea tehnicii booleene permite trecerea de la un set de căi minimale la un set de tăieturi minimale și invers, printr-un procedeu de complementare în doi pași [RAI80]. Se pornește de la funcția  $\Phi$  asociată căilor minimale și pentru a ușura determinarea lui  $\Phi$  (asociată tăieturilor),  $\Phi$  se descompune în subfuncții disjuncte pe baza unei strategii conservative, corespunzător variabilelor aferente muchiilor incidente în nodul sursă (figura 3.8.3).



Figura 3.8.3.

- În primul pas se calculează un polinom modificat, ai cărui termeni corespund la evenimente disjuncte. Acestea se obțin prin gruparea căilor posibile pe clase reciproc exclusive în conformitate cu stările disjuncte ale muchiilor incidente lui  $s$ , care sunt descrise de:

$$d_k = \bar{x}_1 \bar{x}_2 \dots \bar{x}_{k-1} x_k \quad (3.8.15.a)$$

$$d_{n+1} = \bar{x}_1 \bar{x}_2 \dots \bar{x}_n \quad (3.8.15.b)$$

$$\bigcup_{k=1}^{n+1} d_k = 1 \quad (3.8.15.c)$$

Cu acestea, polinomul de cale modificat se poate scrie:

$$\Phi_M = \bigcup_{k=1}^{n+1} d_k \Phi_k = x_1 \Phi_1 \cup \bar{x}_1 x_2 \Phi_2 \cup \dots \cup \bar{x}_1 \dots \bar{x}_{n-1} x_n \Phi_n \cup \bar{x}_1 \dots \bar{x}_{n-1} \bar{x}_n \Phi_{n+1} \quad (3.5.16)$$

unde subfuncția  $\Phi_k$  corespunde căilor posibile în prezența evenimentului  $d_k$ , adică

$$\Phi_k = \Phi_{|x_1 = \dots = x_{k-1} = 0, x_k = 1} \quad (3.8.17)$$

iar totdeauna  $\Phi_{n+1} = 0$  deoarece  $x_1 = \dots = x_n = 0$  întrerupe toate căile de la  $s$  la  $t$ .

- În pasul al doilea se calculează complementul funcției  $\Phi_M$ , care este dat de relația:

$$\bar{\Phi}_M = \bigcup_{k=1}^{n+1} d_k \bar{\Phi}_k \quad (3.8.18)$$

Deci complementul întregii expresii se obține ușor cu ajutorul complementelor funcțiilor  $\Phi_k$ , pentru care se aplică legea lui De Morgan și se ține cont de eventualele absorbții. Pe baza polinomului invers rezultat,

$$\bar{\Phi}_M = x_1 \bar{\Phi}_1 \cup \bar{x}_1 x_2 \bar{\Phi}_2 \cup \dots \cup \bar{x}_1 \dots \bar{x}_{n-1} x_n \bar{\Phi}_n \cup \bar{x}_1 \dots \bar{x}_{n-1} \bar{x}_n \quad (3.8.19)$$

care exprimă situațiile în care nu are loc comunicarea între nodurile  $s$  și  $t$ , se poate deduce setul de tăieturi minimale, reținând elementele date de variabilele

negate din termeni lui  $\overline{\Phi}_M$ . Funcția de structură, scrisă sub forma polinomului de cale (path polinomului), poate fi de asemenea modificată și după variabilele corespunzătoare muchiilor incidente la nodul destinație. Pentru ca evaluarea să fie cât mai simplă, dacă gradul nodului  $s$  este  $D_1$  și al celui  $t$  este  $D_2$ , atunci pentru  $D_1 < D_2$  (respectiv  $D_2 < D_1$ ) se vor selecta pentru modificarea polinomului variabilele corespunzătoare nodului sursă (respectiv destinație).

**Exemplul 3.8.5.** Pentru rețeaua punte din figura 3.7.1, care are funcția de cale dată de (3.7.5), descompunerea după muchiile  $e_1$  și  $e_2$  ( $n=2$ ) dă:

$$\begin{aligned}\Phi_1 &= x_4 \cup x_2 x_5 \cup x_3 x_5 \\ \Phi_2 &= x_5 \cup x_3 x_4 \\ \Phi_3 &= 0\end{aligned}$$

Complementarea acestor funcții este vizibil mai simplă decât a întregii funcții  $\Phi$ :

$$\begin{aligned}\overline{\Phi}_1 &= \overline{x}_4 (\overline{x}_2 \cup \overline{x}_5) (\overline{x}_3 \cup \overline{x}_5) = \overline{x}_2 \overline{x}_3 \overline{x}_4 \cup \overline{x}_4 \overline{x}_5 \\ \overline{\Phi}_2 &= \overline{x}_5 (\overline{x}_3 \cup \overline{x}_4) = \overline{x}_3 \overline{x}_5 \cup \overline{x}_4 \overline{x}_5 \\ \overline{\Phi}_3 &= 1\end{aligned}$$

Setul de tăieturi minimale este dat de variabilele negate din termenii  $\overline{\Phi}_1, \overline{x}_1 \overline{\Phi}_2, \overline{x}_1 \overline{x}_2 \overline{\Phi}_3$ . Considerând după absorbțiile de rigoare, în loc de  $\overline{x}_i \rightarrow e_i$ , se obțin tăieturile minimale:

$$\{e_2, e_3, e_4\}, \{e_4, e_5\}, \{e_1, e_3, e_5\}, \{e_1, e_2\}$$

□

Enumerarea căilor minimale (în care nu se traversează un nod de două ori) și a tăieturilor minimale (care nu conțin o altă tăietură și reduc graful la două subgrafuri conexe), fac obiectul multor cercetări. Astfel, Arunkumar și Lee prezintă un algoritm pentru generarea tuturor tăieturilor minimale dintre o pereche de noduri într-un graf conex neorientat [ARUN79]. Algoritmul evoluează arborescent, până când se obțin două subgrafuri conexe, fiecare conținând unul din cele două noduri care trebuie separate. Partiționarea nodurilor pe nivele se face pe bază de adiacență, începând cu nodul sursă. Totuși, [SANK93] aduce critici acestui algoritm, demonstrând pe un exemplu că pierde anumite tăieturi și concluzionează că el nu merită să fie corectat, deoarece există unul mai bun elaborat de Tsukiyama.

O altă abordare reduce numărul total al căilor minimale, luând în considerare numai căile minimale de bază [JASM85]. *Calea minimală de bază* (basic minimal path) este definită ca și aceea cale minimală în care nici un nod de pe cale nu este legat direct (prin-o singură muchie) de un alt nod al căii, cu excepția celui precedent și al celui următor. De exemplu, pentru graful din figura 3.8.4, căile minimale de bază sunt:

$$\begin{aligned}7-4-2-1 \\ 7-5-2-1 \\ 7-4-3-1 \\ 7-6-3-1\end{aligned}$$

Numerotarea nodurilor sugerează faptul că procedura pentru deducerea acestor căi se bazează pe analiza arborelui de căi minimale care se construiește având în vârf nodul destinație. Abordarea cu ajutorul căilor de bază poate fi privită ca o tehnică de reducere, deoarece combinațiile de defecte care întrerup setul de căi minimale de bază sunt suficiente pentru a determina toate tăieturile minimale ale grafului. Tehnica tradițională determină toate căile minimale, iar tăieturile sunt date de toate combinațiile care întrerup aceste căi. Se demonstrează că setul de tăieturi minimale care deconectează toate căile de bază duce și la deconectarea tuturor celorlalte căi suplimentare (minimale care nu sunt de bază). Pentru cazul considerat, setul de tăieturi minimale de nod este:

Ordinul I : 7, 1

Ordinul II : 3-2

Ordinul III : 6-2-4, 6-4-5, 3-4-5

Metoda permite o reducere considerabilă a efortului de calcul în comparație cu cea clasică, chiar în exemplul dat în loc de studierea totalului de 16 căi minimale a fost suficientă luarea în considerare a 4 căi de bază. Au fost câteva observații în [PRAS94] la algoritmul lui Jasmon de determinare a căilor minimale de bază bazat pe descompunerea grafului în două subgrafuri conexe corespunzătoare nodului sursă și destinație, dar algoritmul a fost corectat de autorul său [JASM94].

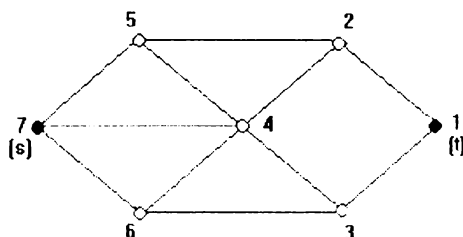


Figura 3.8.4.

Majoritatea algoritmilor de calcul a fiabilității de tip terminal lucrează în două faze: prima dată generează setul de căi sau tăieturi minimale și pe bază lor obțin termenii corespunzători evenimentelor disjuncte care permit scrierea ușoară a funcției de fiabilitate. Există însă și algoritmi care calculează într-o singură etapă funcția de fiabilitate, nefiind necesară cunoașterea prealabilă a informațiilor despre tăieturi sau căi, metoda fiind bazată pe descompunerea matricii de conexiune [RAI86a]. Dezavantajul tuturor acestor procedee constă în efortul mare de calcul necesitat de analiza globală a rețelei, aplicabilitatea lor fiind limitată îndeosebi la rețelele de dimensiuni relativ mici. O abordare alternativă, care reduce efortul de calcul la tratarea pe felii a rețelei, fiind utilizabilă și la sisteme mari, este metoda "holografică" [BENJ86]. Denumirea se datorează faptului că spațiul dintre nodurile terminale  $s$  și  $t$  se particionează pe suprafețe disjuncte, astfel încât:

- fiecare nod se află pe o suprafață
- fiecare muchie unește două suprafețe consecutive sau se află așezate pe o suprafață

Calculul funcției de fiabilitate se face pornind de la nodul  $v_1$  și mergând suprafață cu suprafață, furnizând pentru fiecare suprafață  $S$  informație suficientă privind comunicarea de la  $v_1$  la  $S$  pentru a permite deducerea de informații echivalente referitoare la suprafața  $S+1$ . Suprafețele sunt definite pur topologic, astfel încât suprafața  $S+1$  conține nodurile care pot fi ajunse cu o singură mchie din suprafața  $S$  (figura 3.8.5). Pentru a crea suprafețe complete, se includ și noduri suplimentare perfecte ("dummy" nodes), cum pot fi considerate nodurile  $v_5$  și  $v_6$  de pe figura dată ca exemplu. Se poate arăta că următoarele informații sunt necesare și suficiente pentru a defini influența întregii rețele dinainte de  $S$  și a permite generarea de informații similare în suprafața  $S+1$ :

- pentru o suprafață cu  $N$  noduri, probabilitățile de apariție ale celor  $2^N$  situații diferite de supraviețuire a nodurilor din cadrul suprafeței
- pentru fiecare situație de  $X$  noduri care supraviețuiesc din  $N$ , probabilitățile de apariție ale celor  $2^X$  situații posibile de conectivitate de traversare de la  $v_1$
- pentru fiecare situație de  $Y$  noduri din  $X$  care realizează conectivitatea de traversare, toate cele  $Z$  situații posibile de conectivitate între  $X-Y$  noduri care supraviețuiesc, dar nu sunt noduri de traversare, precum și probabilitățile lor de apariție

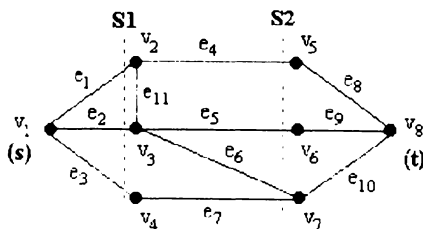


Figura 3.8.5.

De exemplu, pentru suprafața  $S1$  din figură, există următoarele situații distincte de supraviețuire a nodurilor:

$$\begin{array}{ll}
 X=3 & V_2^1 V_3^1 V_4^1 \\
 X=2 & V_2^1 V_3^1 \bar{V}_4^1, V_2^1 \bar{V}_3^1 V_4^1, \bar{V}_2^1 V_3^1 V_4^1 \\
 X=1 & V_2^1 \bar{V}_3^1 \bar{V}_4^1, \bar{V}_2^1 V_3^1 \bar{V}_4^1, \bar{V}_2^1 \bar{V}_3^1 V_4^1 \\
 X=0 & \bar{V}_2^1 \bar{V}_3^1 \bar{V}_4^1
 \end{array}$$

unde  $V_i^1$  reprezintă evenimentul ca nodul  $v_i$  să fie bun, iar  $\bar{V}_i^1$  evenimentul ca să fie defect. Fiecare situație trebuie analizată în funcție de sub-situațiile favorabile, care realizează conexiunea de traversare de la  $v_1$ . De exemplu, probabilitatea ca să existe o conexiune de traversare către  $v_2$  și  $v_3$ , dar nu și  $v_4$  (corespunzător situației  $V_2^1 V_3^1 \bar{V}_4^1$ ), este:

$$\begin{aligned}
 \Pr(V_2^1 V_3^1) = & \left[ p_{v_1} p_1 p_2 + p_{v_1} p_n (p_1 q_2 + q_1 p_2) \right] p_{v_2} p_{v_3} q_{v_4} + \\
 & \left[ p_{v_1} p_1 p_2 q_3 + p_{v_1} p_n q_3 (p_1 q_2 + q_1 p_2) \right] p_{v_2} p_{v_3} p_{v_4}
 \end{aligned} \quad (3.8.20)$$



Probabilitățile corespunzătoare celor șapte situații favorabile de conectivitate pentru suprafața  $S1$  se obțin prin însumarea contribuțiilor corespunzătoare rezultate din analiza fiecărei situații date de evenimentele  $X$ , acestea fiind disjuncte. Probabilitatea finală a conexiunii de traversare prin întreaga rețea este dată de:

$$\begin{aligned} R_{\pi} = \Pr(v_1 \rightarrow v_8) = & \Pr(V_3 V_6 V_7)(1 - q_8 q_9 q_{10}) + \Pr(V_5) p_8 + \\ & + \Pr(V_6) p_9 + \Pr(V_7) p_{10} + \Pr(V_5 V_6)(1 - q_8 q_9) + \\ & + \Pr(V_5 V_7)(1 - q_8 q_{10}) + \Pr(V_6 V_7)(1 - q_9 q_{10}) \end{aligned} \quad (3.8.21)$$

Avantajul metodei se pune în evidență începând cu rețele care au peste cinci suprafețe, efortul de calcul crescând numai liniar cu numărul de suprafețe.

În cazul mai general al problemei conectivității  $(s, T)$  unde  $T \subseteq V - \{s\}$  este o mulțime de noduri destinație, căile minimale sunt date de arborii  $(s, T)$  ai lui  $G$ , adică seturile minime de muchii care conțin o cale de la  $s$  la toate nodurile din  $T$ . În acest caz dacă se cunosc funcțiile de structură corespunzătoare conectivităților  $(s, t_i)$ , unde  $T = \{t_i\}$ , atunci funcția de structură corespunzătoare problemei  $(s, T)$  se va scrie (figura 3.8.6):

$$\Phi_{s,T} = \Phi_{s,t_1} \cap \Phi_{s,t_2} \cap \dots \cap \Phi_{s,t_k} \quad (3.8.22)$$

Termenii acestei funcții, scrisă sub formă minimă disjunctivă, vor reprezenta tocmai setul de arbori minimali care servesc pentru evaluarea probabilității  $(s, T)$ .

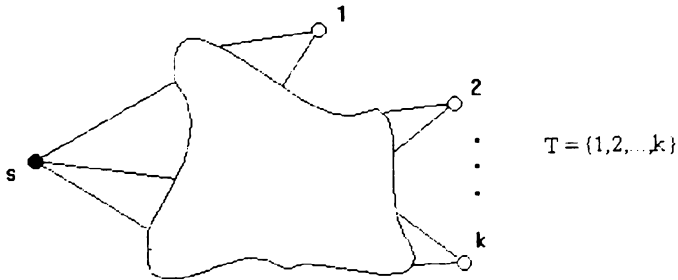


Figura 3.8.6.

**Exemplul 3.8.6.** Dacă la rețeaua punte din figura 3.7.1 se ia  $T = \{u, t\}$ , atunci funcția de structură corespunzătoare conectivității  $(s, T)$  va fi:

$$\begin{aligned} \Phi_{s,T} = \Phi_{s,u} \cap \Phi_{s,t} = & (x_1 \cup x_2 x_3 \cup x_4 x_5)(x_1 x_4 \cup x_2 x_5 \cup x_1 x_3 x_5 \cup x_2 x_3 x_4) = \\ = & x_1 x_4 \cup x_1 x_2 x_5 \cup x_2 x_3 x_5 \cup x_2 x_4 x_5 \cup x_1 x_3 x_5 \cup x_2 x_3 x_4 \end{aligned} \quad (3.8.23)$$

Rezultă imediat seturile de "căi" minimale (de fapt arbori minimali) ca fiind:  $\{e_1, e_4\}$ ,  $\{e_1, e_2, e_5\}$ ,  $\{e_2, e_3, e_5\}$ ,  $\{e_2, e_4, e_5\}$ ,  $\{e_1, e_3, e_5\}$ ,  $\{e_2, e_3, e_4\}$ .

Dacă  $G$  este neorientat, arborii  $(s, T)$  se mai numesc și arbori  $K$  ( $K$ -trees). În situația în care  $T = V - \{s\}$ , arborii  $(s, T)$  formează arbori de acoperire ai grafului cu

rădăcina în  $s$  ( $s$ -rooted spanning arborescences), care la grafuri neorientate devin arbori de acoperire ai lui  $G$  (spanning trees). În cazul în care se pune problema ca fiecare nod din submulțimea  $K \subseteq V$  să poată comunica cu toate celelalte din această submulțime, pentru a calcula fiabilitatea de tip  $K$ -terminal a rețelei, funcția de structură se poate exprima dacă se cunosc drumurile dintre toate perechile de noduri din  $K$ , dar acest lucru nu este necesar, așa cum rezultă din teorema următoare [SATY81][PROV84], dată cu o demonstrație proprie:

**Teorema 3.8.7.** Fiind dat un sistem modelat printr-un graf probabilistic neorientat  $G$  cu  $n$  noduri, funcția de structură corespunzătoare conectivității totale a unei submulțimi  $K$  de noduri (toate nodurile din  $K$  comunică între ele) se calculează pe baza drumurilor dintre un nod oarecare ales sursă ( $s \in K$ ) și celelalte noduri, conform relației

$$\Phi = \Phi_{s1} \cap \Phi_{s2} \cap \dots \cap \Phi_{s(k-1)} \quad (3.8.24)$$

unde  $\Phi_{si}$ ,  $i=1,2,\dots,k-1$ , reprezintă funcția care exprimă conectivitatea între nodul  $s$  și  $i$  ( $s \in K$ ,  $s \neq i$ ), iar  $|K|=k$ .

*Demonstrație.* Este suficient să demonstrez pentru trei noduri ale lui  $G$ , și anume  $K=\{1,2,3\}$ , generalizarea fiind evidentă (figura 3.8.7).

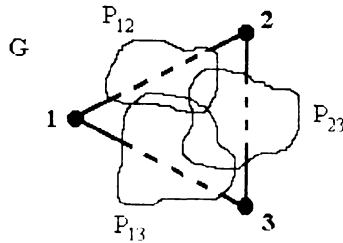


Figura 3.8.7.

Am notat cu  $P_{12}$  reuniunea drumurilor dintre nodurile 1 și 2 care nu trec prin 3 ( $P_{12} = \bigcup_n P_{12_n}$ ), respectiv similar pentru  $P_{13}$  și  $P_{23}$ . Funcția corespunzătoare conectivității totale între cele trei noduri este:

$$\Phi = \Phi_{12} \cap \Phi_{13} \cap \Phi_{23} \quad (3.8.25)$$

unde  $\Phi_{ij} = 1$  dacă există o cale între  $i$  și  $j$ . Aceste funcții se exprimă pe baza figurii ca fiind:

$$\Phi_{12} = P_{12} \cup (P_{13} \cap P_{23}) \quad (3.8.26.a)$$

$$\Phi_{13} = P_{13} \cup (P_{12} \cap P_{23}) \quad (3.8.26.b)$$

$$\Phi_{23} = P_{23} \cup (P_{13} \cap P_{12}) \quad (3.8.26.c)$$

Înlocuind în (3.8.25), dacă se lasă un termen redundant după efectuarea calculelor și se grupează convenabil termenii după nodul 1, se observă că termenii corespunzători lui  $\Phi_{23}$  se absorb:

$$\begin{aligned}\Phi &= [(P_{12} \cap P_{13}) \cup (P_{13} \cap P_{23})] \cup [(P_{12} \cap P_{23}) \cup (P_{12} \cap P_{13} \cap P_{23})] = \\ &= P_{13} \cap [P_{12} \cup (P_{13} \cap P_{23})] \cup (P_{12} \cap P_{23}) [P_{12} \cup (P_{13} \cap P_{23})] = \\ &= P_{12} \cup (P_{13} \cap P_{23}) \cap [P_{13} \cup (P_{12} \cap P_{23})] = \Phi_{12} \cap \Phi_{13}\end{aligned}\quad (3.8.27)$$

□

Pentru a avea un număr minim de căi și deci termeni implicați în calcule, recomand în mod euristic alegerea nodului cu grad de conectivitate maximă drept nod sursă.

**Exemplul 3.8.8.** Alegând la rețeaua punte din figura 3.7.1 nodul  $v$  ca și sursă, funcția de structură pentru calculul fiabilității globale se poate scrie pe baza căilor minimale dintre acest nod și celelalte, ca:

$$\begin{aligned}\Phi &= (x_2 \cup x_1 x_3 \cup x_1 x_4 x_5)(x_1 x_2 \cup x_3 \cup x_4 x_5)(x_5 \cup x_3 x_4 \cup x_1 x_2 x_4) = \\ &= x_1 x_2 x_5 \cup x_2 x_3 x_5 \cup x_1 x_3 x_5 \cup x_2 x_4 x_5 \cup x_1 x_4 x_5 \cup x_2 x_3 x_4 \cup x_1 x_3 x_4 \cup \\ &\cup x_1 x_2 x_4\end{aligned}\quad (3.8.28)$$

Se remarcă faptul că termenii funcției  $\Phi$  reprezintă tocmai arborii de acoperire minimali ai grafului. Dacă se alegea nodul  $t$  (de grad 2) ca și sursă, trebuie operat cu  $4+3+3=10$  căi în loc de 9, ca pentru situația considerată.

□

Seturile de tăieturi pentru problema conectivității ( $s, T$ ), sunt constituite din tăieturile ( $s, T$ ) ale lui  $G$ , adică mulțimile minimale de muchii a căror eliminare deconectează pe  $s$  de niște noduri din  $T$ . Dacă  $T = V - \{s\}$ , aceste tăieturi se numesc tăieturile de rețea (network cuts) ale lui  $G$ , orientate după  $s$  ( $s$ -directed) dacă  $G$  este de tip orientat.

**Limite de fiabilitate.** În practică pot fi întâlnite sisteme complexe ale căror structuri nu sunt serie-paralel și nu pot fi reduse, sau datorită numărului mare de componente calculul exact al fiabilității nu este convenabil. În asemenea cazuri devin folositoare *metodele aproximative*. Limite grosolane sunt obținabile prin determinarea fiabilității sistemelor formate prin conectarea în serie a tuturor componentelor din sistemul studiat (pentru limita inferioară), respectiv prin conectarea lor în paralel (pentru limita superioară). O asemenea pereche de limite este grosolană deoarece nu ia în considerare structura existentă a sistemului. Limite mai bune pot fi obținute prin utilizarea conceptelor de căi minimale și tăieturi minimale, pe baza teoremei următoare [VISW81]:

**Teorema 3.8.9.** Fie  $\Phi(N)$  funcția de structură a unui sistem coerent și  $R(p_1, p_2, \dots, p_n)$  funcția sa de fiabilitate. Dacă  $C_1, C_2, \dots, C_m$  reprezintă tăieturile minimale ale sistemului și  $P_1, P_2, \dots, P_l$  căile minimale ale acestuia, atunci:

$$\prod_{i=1}^m \left[ 1 - \prod_{j \in C_i} (1 - p_j) \right] \leq R(p_1, p_2, \dots, p_n) \leq 1 - \prod_{k=1}^l \left[ 1 - \prod_{j \in P_k} p_j \right] \quad (3.8.29)$$

*Demonstrație.* Asociind fiecărei tăieturi  $C_i$  câte o funcție ajutătoare crescătoare

$$Z_i(X) = 1 - \prod_{c_j \in C_i} (1 - x_j)$$

și ținând cont de faptul că covarianța a două funcții aleatoare crescătoare este pozitivă [TĂRC89], adică  $E[Z_i(X)Z_{i+1}(X)] \geq E[Z_i(X)]E[Z_{i+1}(X)]$  pe baza relației (3.5.4) și (3.8.9) se poate scrie:

$$\begin{aligned} R(p_1, p_2, \dots, p_n) &= E[\Phi(X)] = E\left[\prod_{i=1}^n Z_i(X)\right] \geq \prod_{i=1}^n E[Z_i(X)] = \\ &= \prod_{i=1}^m \Pr(Z_i(X) = 1) = \prod_{i=1}^m \left[1 - \prod_{c_j \in C_i} (1 - p_j)\right] \end{aligned} \quad (3.8.30)$$

În mod similar fiecărei căi  $P_k$  i se asociază câte o funcție crescătoare

$$y_k(X) = \prod_{c_j \in P_k} x_j$$

cu ajutorul cărora se poate scrie:

$$\begin{aligned} 1 - R(p_1, p_2, \dots, p_n) &= E[1 - \Phi(X)] = E\left[\prod_{k=1}^l (1 - y_k(X))\right] \geq \prod_{k=1}^l E[1 - y_k(X)] = \\ &= \prod_{k=1}^l [1 - \Pr(y_k(X) = 1)] = \prod_{k=1}^l \left[1 - \prod_{c_j \in P_k} p_j\right] \end{aligned} \quad (3.8.31)$$

Relațiile (3.8.30) și (3.8.31) corespund inegalității din enunț. □

Pe baza căilor și tăieturilor se construiesc două sisteme auxiliare în vederea evaluării limitelor de fiabilitate (reliability bounds). Primul sistem conține configurația paralelă a componentelor inseriate pentru fiecare cale minimală, iar al doilea este compus din legarea în serie a tuturor conexiunilor paralele ale componentelor pentru fiecare tăietură minimală.

**Exemplul 3.8.10.** Pentru calculul limitelor fiabilității de tip terminal între perechea de noduri  $(s, t)$  a rețelei punte din figura 3.7.1, cele două subsisteme auxiliare sunt reprezentate de schemele bloc de fiabilitate din figura 3.8.8, obținute pe baza tabelului 3.8.2 de căi și tăieturi minimale. Sistemul  $S_1$  furnizează, conform teoremei 3.8.9, o limită superioară (upper bound) a fiabilității sistemului original, care se poate calcula ușor pe baza compunerilor serie și paralel, ca fiind:

$$R_{\text{sup}} = 1 - (1 - p_1 p_4)(1 - p_2 p_5)(1 - p_1 p_3 p_5)(1 - p_2 p_3 p_4) \quad (3.8.32)$$

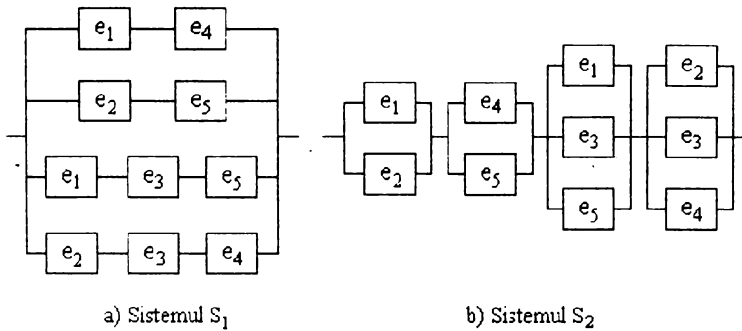


Figura 3.8.8.

Analog se obține o limită inferioară (lower bound) a fiabilității sistemului pe baza schemei bloc  $S_2$ :

$$R_{\text{inf}} = [1 - (1 - p_1)(1 - p_2)] [1 - (1 - p_4)(1 - p_5)] [1 - (1 - p_1)(1 - p_3)(1 - p_5)] \cdot [1 - (1 - p_2)(1 - p_3)(1 - p_2)] \quad (3.8.33)$$

Dacă se consideră componentele de aceeași fiabilitate  $p=0,9$ , atunci limitele de fiabilitate sunt  $R_{\text{sup}} = 0,99734$  și  $R_{\text{inf}} = 0,97814$ , față de fiabilitatea  $R_{\pi} = 0,97848$  a sistemului conform relației exacte (3.7.6). □

Cum se poate observa și din exemplul precedent, limitele de fiabilitate date de (3.8.29) sunt destul de largi (pentru media aritmetică a limitelor  $R_{\text{med}} = 0,98774$ , nefiabilitatea prezintă o abatere relativă de 43% față de valoarea exactă), de aceea s-au căutat și alte tehnici de aproximare pornind de la seturile de căi și tăieturi minimale. Una dintre ele pornește de la enumerarea tuturor căilor (tăieturilor) minimale și trunchierea expresiei de fiabilitate. Calculând numai primii  $k$  termeni ai dezvoltării, se obține o limită superioară (inferioară), metoda fiind numită *aproximare prin trunchiere* (truncation approximation). Pentru determinarea ambelor limite se poate lucra și numai cu mulțimea tăieturilor minimale dacă pentru expandarea relației de probabilitate (3.8.7) se folosește formula de incluziune-excluziune (3.7.4). Termenii acestei formule scad ca mărime, cei de ordinul întâi fiind puteri mici ale lui  $q$  în funcție de numărul de muchii din tăieturi, cei de ordinul al doilea fiind de fapt produse ale celor de ordinul întâi, ș.a.m.d.. Deoarece termenii dezvoltării alternează ca semn, se obțin niște limite alternante, cu atât mai precise cu cât numărul de termeni luați în calcul este mai mare. Reținând numai termenii de ordinul întâi ai dezvoltării reuniunii de tăieturi, se obține o limită inferioară a fiabilității. Însurmand la aceasta și termenii de ordinul al doilea rezultă o limită superioară de fiabilitate. Valoarea aproximativă a fiabilității se poate lua ca media celor două limite.

**Exemplul 3.8.11.** Expresia de fiabilitate a conectivității nodurilor ( $s,t$ ) a rețelei punte (figura 3.7.1) pe baza setului de tăieturi minimale este, conform (3.8.7):

$$\begin{aligned}
 R_{\pi} = 1 - \{ & [\Pr(\bar{x}_1\bar{x}_2) + \Pr(\bar{x}_4\bar{x}_5) + \Pr(\bar{x}_1\bar{x}_3\bar{x}_5) + \Pr(\bar{x}_2\bar{x}_3\bar{x}_4)] - [\Pr(\bar{x}_1\bar{x}_2\bar{x}_4\bar{x}_5) + \\
 & + \Pr(\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_5) + \Pr(\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4) + \Pr(\bar{x}_1\bar{x}_3\bar{x}_4\bar{x}_5) + \Pr(\bar{x}_2\bar{x}_3\bar{x}_4\bar{x}_5) + \\
 & + \Pr(\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4\bar{x}_5)] + [4 \Pr(\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4\bar{x}_5)] - [\Pr(\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4\bar{x}_5)] \} \quad (3.8.34)
 \end{aligned}$$

Dacă toate muchiile au aceeași probabilitate de defectare  $q$ , relația devine:

$$R_{\pi} = 1 - \{ [2q^2 + 2q^3] - [5q^4 + q^5] + [4q^5] - [q^5] \} \quad (3.8.35)$$

Pentru  $p=0,9$ , deci  $q=1-p=0,1$ , fiabilitatea va fi:

$$R_{\pi} = 1 - \{ [0,022] - [0,00051] + [0,00004] - [0,00001] \} \quad (3.8.36)$$

Prin adunarea pe rând a valorilor din paranteze se obține o mulțime alternantă de limite:

$$\begin{aligned}
 R_{\pi} &< 1 \\
 R_{\pi} &> 0,978 \\
 R_{\pi} &< 0,97851 \\
 R_{\pi} &> 0,97847 \\
 R_{\pi} &= 0,97848
 \end{aligned}$$

În acest caz media limitelor de ordinul unu și doi dau  $R_{med}=0,978255$ , ceea ce reprezintă o eroare relativă a nefiabilității față de valoarea exactă de numai 1,04%. □

Un alt procedeu de aproximare implică aflarea numai a unei submulțimi a căilor sau tăieturilor minimale, care sunt cel mai ușor de găsit și contribuie cel mai mult la fiabilitatea rețelei. De regulă se iau în considerare căile sau tăieturile care conțin un număr minim de muchii (numite și "hopuri"), metoda fiind numită aproximarea cu hopuri minime (min hop approximation). Astfel se reduce substanțial dificultatea legată de găsirea căilor și tăieturilor, dar pentru determinarea limitelor trebuie efectuat calculul asupra tuturor termenilor din dezvoltarea formulei de reuniune a probabilităților. Căile minimale alese vor furniza o limită inferioară a fiabilității, iar tăieturile minimale o limită superioară. Există algoritmi cu un parametru de intrare  $k$ , care furnizează valoarea aproximativă a fiabilității pe baza a  $k$  căi care conțin un număr minimal de muchii (the  $k$ -shortest tie-set method) [MURR93]. Cele mai fiabile  $k$  căi (respectiv tăieturi) se aleg pe baze euristice, astfel încât probabilitatea reuniunii acestora să maximalizeze (respectiv să minimalizeze) limita inferioară (superioară). Cele mai convenabile căi și tăieturi din acest punct de vedere sunt cele care nu conțin elemente comune și au cardinalitatea minimă. S-a arătat experimental că deși teoretic exponențială, metoda s-a dovedit a fi polinomială în practică în raport cu numărul de căi și tăieturi.

Abordarea limitelor fiabilității de tip  $(s,t)$  se poate face și pe baza polinomului de fiabilitate în  $p$ , considerând componentele identice. Acest polinom are forma generală:

$$R_{\pi} = \sum_{i=0}^n f_i p^{n-1} (1-p)^i \quad (3.8.37)$$

unde  $n$  reprezintă numărul de muchii, iar  $f_i$  numărul de căi complementare diferite de dimensiune  $i$ . Călea complementară (co-pathset) este o submulțime de componente al cărei complement este o cale (nu minimală), adică defectarea unei căi complementare lasă sistemul funcțional. Polinomul de fiabilitate transformă problema fiabilității în problema de determinare a celor  $m+1$  coeficienți din (3.8.37). Unii coeficienți  $f_i$  se pot afla ușor. Dacă se notează cu  $c$  cardinalitatea celei mai scurte tăieturi  $s$ - $t$ , coeficienții  $f_i$  pentru  $i < c$  se determină pe baza următoarei observații: orice mulțime de muchii de dimensiune  $i < c$  nu poate fi o tăietură, deci constituie o cale complementară. Numărul total al acestora este  $\binom{n}{i}$ , deci  $f_i = \binom{n}{i}$  pentru  $i < c$ . Fie acum  $d$  numărul maxim de muchii care pot fi eliminate din  $G$  fără a deconecta legătura dintre  $s$  și  $t$ . Diferența  $n-d$  va reprezenta dimensiunea celei mai scurte căi dintre  $s$  și  $t$ , deci numărul de modalități în care  $d$  muchii pot fi eliminate fără deconectarea lui  $G$  este egal cu numărul celor mai scurte căi  $s$ - $t$  din  $G$ . Acesta este tocmai  $f_d$ , și poate fi calculat în timp polinomial. Dacă  $i > d$  atunci  $f_i = 0$ , deoarece în cazul defectării a mai mult de  $d$  muchii  $s$  și  $t$  vor fi sigur deconectate, întrucât numărul muchiilor operaționale este mai mic decât dimensiunea celei mai scurte căi  $s$ - $t$  din  $G$ . În consecință, coeficienții necunoscuți  $f_i$  din relația (3.8.37) sunt cei pentru  $c \leq i < d$ :

$$\underbrace{f_0, \dots, f_{c-1}}_{f_i = \binom{n}{i}} \underbrace{f_c, \dots, f_{d-1}}_{\text{necunoscuți}} f_d, \underbrace{f_{d+1}, \dots, f_m}_0$$

Prin aproximarea coeficienților necunoscuți dintre  $f_c$  și  $f_{d-1}$  în (3.8.37) se poate obține o aproximare a fiabilității aferente conectivității  $(s, t)$ . În [PROV86] se transpune metoda pentru cazul fiabilității globale, în locul coeficienților necunoscuți corespunzătorii limitelor de fiabilitate fiind introdusă noțiunea de  $h$ -vector, care se evaluează pe criterii combinatoriale. Pentru cazul polinomului de fiabilitate cu două variabile (muchii și noduri expuse la defectare, cu probabilități diferite) devine necesară stabilirea coeficienților  $f_i$  ai căte unei matrici pentru limita inferioară, respectiv superioară [BUJKA94].

Problema limitelor de fiabilitate se poate pune și sub următoarea formă: știind că fiabilitățile componentelor variază între anumite limite date, care sunt cele mai bune limite inferioare și superioare pentru fiabilitatea sistemului? Soluția este dată de rezolvarea unei probleme de programare liniară pornind de la setul de căi minimale  $\mathcal{P}$  și setul de tăieturi minimale  $\mathcal{C}$  dintre nodurile  $s$ - $t$  [ASSO86]. Dacă pentru fiecare componentă  $c_i$  este dată informația

$$a_i \leq \Pr\{\text{componenta } c_i \text{ operațională}\} \leq b_i$$

atunci cea mai bună limită superioară a fiabilității de tip terminal este dată de

$$R_{\max} = \min \left\{ 1, \min_{\mathcal{C} \in \mathcal{C}} \sum_{c_i \in \mathcal{C}} b_i \right\} \quad (3.8.38)$$

iar cea mai bună limită inferioară este:

$$R_{\min} = \max \left\{ 0, 1 - \min_{\mathcal{P} \in \mathcal{P}} \sum_{c_i \in \mathcal{P}} (1 - a_i) \right\} \quad (3.8.39)$$

Demonstrația se bazează pe teorema lui Ford și Fulkerson, care afirmă că fluxul maxim între nodurile  $s$  și  $t$  este dat de capacitatea tăieturii minime (max-flow min-cut) [MAYE72]. Pentru cazul limitei superioare de fiabilitate, capacitatea tăieturii minime (prin capacitate înțelegându-se valoarea  $b$ , asignată muchiei) determină o împachetare optimă a căilor (edge-packing) din punct de vedere al fiabilității. La fel, pentru limita inferioară, calea minimă (cea mai scurtă) determină o împachetare maximă a tăieturilor (capacitățile asignate în acest caz muchiilor fiind valorile  $a_i$ ).

Determinarea setului de căi și de tăieturi minimale, precum și a fiabilității estimate a unei rețele, se poate face printr-o *simulare Monte Carlo* [LIN93]. Desigur prin această tehnică se obține o valoare numerică a fiabilității, fiind dat parametrul distribuției pentru fiecare componentă (valorile  $\lambda_i$  pentru distribuția exponențială) și schema bloc de fiabilitate a sistemului. Prima dată se determină setul de căi minimale prin parcurgerea sistemului de la  $s$  la  $t$  într-o manieră aleatoare. Se generează o cale aleatoare trecând prin succesorul fiecărei componente, pornind de la sursă și până când se atinge destinația. Dacă o componentă are mai mulți succesori, atunci se alege unul dintre ei în mod aleator, cu probabilități egale. În situația în care succesorul selectat face parte deja din calea curentă, atunci înseamnă că s-a format o buclă, se renunță la acesta și se alege în mod aleator un alt succesor dintre cei rămași. La un algoritm de explorare exhaustivă, deoarece procesul aleator nu garantează căi unice, fiecare cale găsită se compară cu cele anterioare pentru a elimina redundanța. Aplicând teorema absorbției, se reține calea care conține numărul minim de componente. Pentru a asigura ca mulțimea de căi minimale să fie completă, procesul se repetă până când de 500 de ori nu se mai găsește o cale nouă [LIN93]. Setul de căi minimale se folosește în continuare ca și criteriu de decizie asupra nefuncționalității sistemului, acesta fiind considerat defect dacă după căderea unui număr de componente nu a rămas nici o cale operațională. Pentru fiecare componentă se generează aleator un timp până la defectare, conform distribuției sale. Se face lista timpilor de defectare în ordinea producerii și se elimină pe rând componentele defecte până când se constată nefuncționalitatea sistemului. După defectarea fiecărei componente, cele rămase se verifică față de căile minimale și dacă se constată că sistemul nu este funcțional, atunci înseamnă că componentele defecte formează o tăietură. Tăieturile minimale se determină pe baza acestora folosind teorema absorbției, ca la căile minimale. Pentru fiecare simulare, care trebuie să fie de ordinul sutelor pentru a se obține rezultate acceptabile, se obține câte un timp de defectare al sistemului, media lor fiind valoarea medie a timpului de funcționare (*MTTF*). Curba de fiabilitate a sistemului se determină prin distribuirea timpilor de defectare ai sistemului pe clase corespunzătoare la intervale egale de timp. Astfel se pot calcula ratele de defectare pentru fiecare interval și valoarea de nefiabilitate cumulată a sistemului la aceste momente fixe de timp. Valoarea distribuției cumulate a fiabilității sistemului se obține scăzând din 1 valorile de nefiabilitate cumulate, iar între două momente de timp considerate se poate aplica o interpolare. O rezoluție mai bună se obține alegând mai multe intervale, ceea ce presupune însă cât mai multe date (timpuri de defectare ai sistemului), care se pot obține în urma unui număr mare de rulări de simulare (simulation runs).

De multe ori se poate întâmpla, la sistemele mari, ca tehnicile analitice prezentate, chiar pentru încadrarea fiabilității sistemului între anumite limite, să devină deosebit de complexe. În acest caz sunt adecvate tehnicile de simulare (simulation techniques), cu toate că ele produc numai estimări și nu rezultate exacte [BARBU92]. Este necesar un număr mare de rulări pentru a reduce varianța și a obține indicatori acceptabili.



### 3.9. Utilizarea tehnicii booleene pentru calculul fiabilității

În cele prezentate până acum, schemele bloc de fiabilitate ale sistemelor complexe au fost analizate pe baza formulelor de reducere de tip serie-parallel sau a formulelor de probabilitate aplicate asupra expresiei analitice a funcției de structură. O metodă alternativă, menită să simplifice calculele algebrice laborioase din domeniul probabilistic, constă în a considera parametrii de fiabilitate ale componentelor ca fiind variabile booleene și a muta problema de calcul în domeniul algebrei booleene. Avantajul de seamă îl reprezintă posibilitatea utilizării puternicelor teoreme de reducere booleene (idempotența absorbția etc.), care limitează dimensiunea problemei de calcul. Scopul urmărit este de a putea găsi o funcție booleană de variabile  $x_i$  asociate componentelor, care să poată fi interpretată direct (unu la unu) ca și expresia aritmetică a fiabilității sistemului, făcând substituiriile:

$$x_i \rightarrow p_i \quad (3.9.1.a)$$

$$\bar{x}_i \rightarrow q_i = 1 - p_i \quad (3.9.1.b)$$

$$\text{sumă booleană} \rightarrow \text{sumă algebrică} \quad (3.9.1.c)$$

$$\text{produs boolean} \rightarrow \text{produs algebric} \quad (3.9.1.d)$$

Criteriul	Domeniul boolean	Domeniul probabilistic
1. Diagrama	Karnaugh (K-map)	De probabilitate (P-map)
2. Variabile	Booleene (valori discrete de 1 și 0 corespunzătoare lui adevărat și fals)	Probabilistice (valori continue între 0 și 1 cu valori extreme corespunzătoare situației de sigur defect și sigur funcțional)
3. Formula de calcul	Axiomele și teoremele algebrei booleene	Axiomele și teoremele teoriei probabilităților
4. Operatori	SAU ( $\cup$ ) ȘI ( $\cap$ ) NU ( $\bar{\phantom{x}}$ )	Plus (+) Minus (-) Înmulțire ( $\cdot$ ) Împărțire ( $:$ )

Tabelul 3.9.1.

O paralelă sugestivă a algebrei booleene cu cea probabilistică este dată în tabelul 3.9.1. În [KAKAS94c] se face o analiză a modului cum se reflectă fiecare din metodele de bază pentru calculul fiabilității unui sistem multiterminal, prezentate la paragraful 3.7, în domeniul boolean. Consider o diagramă Karnaugh de 5 variabile, în care fiecare căsuță corespunde unei stări posibile a sistemului (figura 3.9.2). Aplicațiile în "1" denotă funcționalitatea rețelei de tip punte din figura 3.7.1 pentru problema conectivității  $s-t$ . Forma canonică a funcției booleene din diagramă reprezintă tocmai funcția de structură descrisă de (3.7.2). Termenii acesteia, ca de altfel căsuțele diagramei Karnaugh, corespund la evenimente disjuncte ( $A_i \cap A_j = \emptyset$ ), deci funcția de fiabilitate se poate obține imediat, efectuând substituțiile de la (3.9.1):

$$\begin{aligned}
 R_{\pi} = & p_1 p_2 q_3 p_4 q_5 + p_1 q_2 q_3 p_4 q_5 + q_1 p_2 p_3 p_4 q_5 + p_1 p_2 p_3 p_4 q_5 + \\
 & + p_1 q_2 p_3 p_4 q_5 + q_1 p_2 q_3 q_4 p_5 + p_1 p_2 q_3 q_4 p_5 + q_1 p_2 q_3 p_4 p_5 + \\
 & + p_1 p_2 q_3 p_4 p_5 + p_1 q_2 q_3 p_4 p_5 + q_1 p_2 p_3 p_4 p_5 + p_1 p_2 p_3 p_4 p_5 + \\
 & + p_1 q_2 p_3 p_4 p_5 + q_1 p_2 p_3 q_4 p_5 + p_1 p_2 p_3 q_4 p_5 + p_1 q_2 p_3 q_4 p_5
 \end{aligned} \quad (3.9.2)$$

Expresia obținută corespunde formulei de probabilitate (3.7.1) care enumeră toate stările de succes (dezvoltare canonică directă).

		$x_1$		$x_5$					
					$x_1$				
		0	0	0	0	1	1	0	
		0	0	1	1	0	1	1	1
	$x_3$	0	1	1	1	0	1	1	1
		0	0	0	0	0	1	1	1
		$x_2$			$x_2$				

Figura 3.9.2.

Dacă se face minimizarea funcției din diagrama Karnaugh de pe figura 3.9.2, se obține:

$$\Phi = x_1 x_2 \vee x_2 x_5 \vee x_1 x_3 x_5 \vee x_2 x_3 x_4 \quad (3.9.3)$$

Se observă că s-a obținut tocmai funcția de structură sub formă booleană, termenii acesteia reprezentând căile minimale posibile dintre  $s$  și  $t$ . De asemenea este de remarcat faptul că nu există altă modalitate de minimizare a funcției, ceea ce m-a condus la statuarea următoarei teoreme:

**Teorema 3.9.1.** Dacă  $\Phi$  este funcția de structură a unui sistem coerent exprimată sub formă booleană, atunci forma ei minimă disjunctivă este unică și termenii ei corespund setului de căi minimale.

*Demonstrație.* Se pornește de la forma canonică disjunctivă a lui  $\Phi$ , care conține toți termenii canonici corespunzători funcționării corecte a sistemului. Deoarece sistemul este coerent, adică defectarea unei componente ( $\bar{x}$ ) nu poate cauza trecerea sistemului în stare funcțională, fiecărui termen de forma  $Y\bar{x}$  îi corespunde și un termen de forma  $Yx$  în expresia lui  $\Phi$  (invers nu mai este adevărată afirmația). Dacă este așa, atunci  $Y\bar{x} \cup Yx = Y$ , termenii fiind comasați. Aplicând succesiv această proprietate pornind de la forma canonică, dispar toate variabilele negate și termenii rămași corespund căilor (conform definiției 3.8.2). După eliminarea termenilor redundanți, rămân cei corespunzători căilor minimale, care este o mulțime unică a sistemului. De fapt acești termenii reprezintă implicații primi ai funcției  $\Phi$  (înlăturând oricare variabilă dintr-un termen, încetează să mai fie o cale), expresia obținută fiind forma disjunctivă prescurtată [HUȚA83] sau expresia caracteristică [MUNT77]. Această formă se identifică însă aici cu forma disjunctivă minimă, toți implicații primi fiind esențiali. Pentru fiecare implicanț prim de



Expresia (3.9.5) corespunde calculului de fiabilitate utilizând sume de produse disjuncte și termenii săi sunt scriși în corespondență directă cu formula (3.7.10) a probabilităților condiționate. Evident forma disjunctă a funcției nu mai este unică, de aceea fiabilitatea se poate exprima sub forma mai multor expresii echivalente. Scopul este însă de a obține o expresie cât mai simplă cu număr minim de termeni, problemă de care mă voi ocupa separat în paragraful următor. Metoda diagramei de fiabilitate este foarte comodă de aplicat pentru calcule de mână având avantajul clarității, dar numai pentru sistemele cu  $n \leq 6$  componente, pentru care se pot întocmi diagramele de minimizare.

Pentru a utiliza principiul de incluziune-excluziune dat de formula (3.7.4) pentru determinarea expresiei de fiabilitate, trebuie luați în considerare și elementii comuni ai cuburilor (nedisjuncte) din diagrama de minimizare. Astfel, relația de fiabilitate se poate obține pe baza grupării termenilor din figura 3.9.2, prin scăderea din cuburile minimale a termenilor comuni de atâtea ori de câte ori sunt acoperiți mai mult decât o singură dată,

deoarece  $\sum_{i=1}^k \Pr\{A_i\} = k \Pr\{A\}$  și nu  $\Pr\{A\}$ :

$$\begin{aligned} R_{\pi} &= p_1 p_4 + p_2 p_5 + p_1 p_3 p_5 + p_2 p_3 p_4 - (p_1 p_2 p_4 p_5 + p_1 p_3 p_4 p_5 + \\ &\quad - p_1 p_2 p_3 p_4 - p_1 p_2 p_3 p_5 + p_2 p_3 p_4 p_5) + 2 p_1 p_2 p_3 p_4 p_5 = \\ &= p_1 p_4 + p_2 p_5 + p_1 p_3 p_5 + p_2 p_3 p_4 - (p_1 p_2 p_3 p_4 p_5 + p_1 p_2 p_3 p_4 p_5 + \\ &\quad - p_1 p_2 p_3 p_4 p_5 + p_1 p_2 p_3 p_4 p_5 + 3 p_1 p_2 p_3 p_4 p_5) \end{aligned} \quad (3.9.6)$$

Descompunerea după un element cheie (3.7.15) se poate aplica dacă numărul de componente este  $n > 6$ , descompunerea după fiecare componentă dând naștere la câte două diagrame de minimizare.

**Generalizare la fiabilitatea multiterminal.** Performanța de îndeplinire a unei anumite funcții de către un sistem distribuit poate fi caracterizată printr-o fiabilitate de tip multiterminal, exprimată ca probabilitatea de a fi adevărată o funcție booleană, ai căror termeni denotă existența unor conexiuni între subseturi de resurse. Funcția booleană introdusă de Fratta și Montanari pentru reprezentarea conectivității între două noduri se poate scrie relativ ușor la sistemele mici pe baza topologiei. Implicanții acestei funcții se mai numesc identificatori de cale (path identifiers). În [GRNA81] metoda se extinde la calculul fiabilității de tip multiterminal, utilizând operatori ȘI și SAU la descrierea conexiunilor între grupe de noduri. Dacă se lucrează cu noduri imperfecte și pentru acestea se pot introduce variabile în termenii funcției. Pentru determinarea fiabilității corespunzătoare conexiunilor multiterminal descrise de funcția  $\Phi_m$ , trebuie găsită o acoperire minimă cu implicanți disjuncți în vederea transpunerii expresiei într-o sumă de termeni probabilistici de forma

$$\Phi_m = \sum_{\Phi_m(A)=1} P_A \quad (3.9.7)$$

unde  $P_A$  este probabilitatea ca evenimentul  $A$  să se producă ( $\Phi_m(A) = 1$ ), adică să existe conectivitatea între nodurile specificate prin enunț. Odată găsită această formă booleană, expresia aritmetică a fiabilității se obține făcând substituirile (3.9.1).

**Exemplul 3.9.3.** Dacă pentru rețeaua punte din figura 3.7.1 se cere fiabilitatea corespunzătoare conexiunilor multiterminal

$$C_m = C_{s,\mu} \text{ SAU } C_{s,t}$$

se obține funcția booleană

$$\Phi_m = \Phi_{s,\mu} \cup \Phi_{s,t}$$

unde  $\Phi_{s,\mu} = x_1 \cup x_2 x_3 \cup x_2 x_4 x_5$  și  $\Phi_{s,t} = x_1 x_4 \cup x_1 x_3 x_5 \cup x_2 x_5 \cup x_2 x_3 x_4$ . Funcția  $\Phi_m$  este reprezentată pe diagrama Karnaugh din figura 3.9.4. Luând acoperirile cu cuburi disjuncte, se obține pentru  $\Phi_m$  forma

$$\Phi_m = x_1 \cup \bar{x}_1 x_2 x_5 \cup \bar{x}_1 x_2 x_3 \bar{x}_5$$

din care se obține imediat expresia simbolică a fiabilității de tip multiterminal cerute:

$$R_m = p_1 + q_1 p_2 p_5 + q_1 p_2 p_3 p_5$$

□

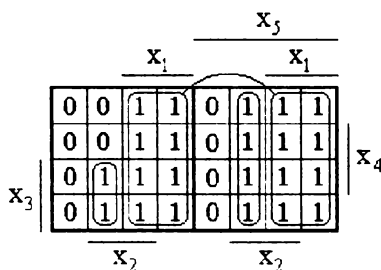


Figura 3.9.4.

Metoda prezentată se poate utiliza cu succes la determinarea probabilității de îndeplinire a unei anumite aplicații într-o rețea, cunoscându-se fiabilitățile nodurilor procesoare și a legăturilor, precum și distribuția programelor și datelor (paragraful 5.1).

### 3.10. Disjunctizarea termenilor din funcția de structură

După cum s-a văzut din analiza făcută în paragraful precedent, forma cea mai simplă a expresiei de fiabilitate, cu număr minim de termeni, se obține prin considerarea cuburilor disjuncte de pe diagrama de minimizare a funcției de structură aferente. Tehnica booleană devine deci comodă de utilizat, inclusiv pentru implementare pe calculator, dacă expresia booleană a acestei funcții se aranjează astfel încât să conțină numai termeni disjuncți înainte de conversia ei în domeniul probabilistic. Evenimentele de succes  $A_i$  asociate termenilor disjuncți, fiind reciproc exclusive, transpunerea în funcția de fiabilitate este posibilă pe baza relației de probabilitate:

$$R = \Pr\left\{\bigcup_{i=1}^m A_i\right\} = \sum_{i=1}^m \Pr\{A_i\} \quad (3.10.1)$$

unde  $m$  reprezintă numărul termenilor disjuncți. Rezultatul obținut va fi o *sumă de produse disjuncte* (SDP, sum of disjoint products). Dacă  $A_i$  reprezintă evenimentele asociate cuburilor disjuncte date de comasarea 0-urilor de pe diagrama de minimizare, atunci suma de produse disjuncte corespunzătoare va da expresia funcției de nefiabilitate  $F = 1 - R$ . Cu această precizare tot ce voi prezenta în continuare referitor la disjunctizarea termenilor este valabil, atât în cazul în care termenii desemnează căi, cât și în cazul în care ei desemnează tăieturi. În primul rând, se pune întrebarea care este condiția ca doi termeni booleani ai unei expresii disjunctive să fie disjuncți? Răspunsul este dat de următoarea teoremă [ABRA79]:

**Teorema 3.10.1.** Doi termeni booleani conjunctivi sunt disjuncți dacă și numai dacă un termen conține o variabilă booleană  $x$  și celălalt conține aceeași variabilă dar negată  $\bar{x}$ .

*Demonstrația* este imediată și se bazează pe faptul că unul din termeni va fi inclus în domeniul corespunzător lui  $x$  de pe diagrama Karnaugh, iar celălalt în domeniul corespunzător lui  $\bar{x}$ , cele două domenii neavând nici un element comun.  $\square$

Dacă se compară doi termeni ai unei sume de produse booleene, atunci pe baza teoremei precedente se poate stabili dacă ei sunt disjuncți sau nu. În situația în care ei nu sunt disjuncți, atunci trebuie efectuată o modificare asupra unuia din termeni. Fie  $T_1$  și  $T_2$  două produse booleene nedisjuncte și  $\mathcal{A} = \{x_1, x_2, \dots, x_k\}$  complementul relativ al lui  $T_1$  față de  $T_2$  (mulțimea variabilelor care există în  $T_1$ , dar nu există în  $T_2$ ). (Toate variabilele  $x_i$  sunt scrise aici sub formă necomplementată, dar acestea nu afectează generalitatea rezultatului). Poate avea loc unul din următoarele cazuri:

1. Dacă  $\mathcal{A} = \emptyset$ , atunci  $T_1 \cap T_2 = T_1$  deoarece termenul  $T_2$  este acoperit de  $T_1$ .
2. Dacă  $\mathcal{A} \neq \emptyset$ , atunci  $T_2$  se face disjunct cu  $T_1$  prin introducerea variabilelor lipsă pe baza următoarei dezvoltări care conține numai termeni reciproc exclusivi:

$$T_1 \cap T_2 = T_1 \cap \bar{x}_1 T_2 \cup x_1 \bar{x}_2 T_2 \cup \dots \cup x_1 x_2 \dots \bar{x}_k T_2 \quad (3.10.2)$$

Această tehnică este general valabilă și se poate utiliza totdeauna când se calculează fiabilitatea (nefiabilitatea) pe baza funcției booleene date de setul de căi (tăieturi) minimale. Astfel ea stă la baza algoritmului lui Abraham [ABRA79], prezentat pe o rețea de referință în Anexa 1.

**Exemplul 3.10.2.** Pentru a ilustra procedeul, aleg funcția corespunzătoare setului de căi minimale  $s-t$  din rețeaua punte (figura 3.7.1):

$$\Phi = x_1 x_4 \cup x_2 x_5 \cup x_1 x_3 x_5 \cup x_2 x_3 x_4 \quad (3.10.3)$$

Selectând primul termen (marcat cu săgeată) și comparându-l cu ceilalți pentru a-i face disjuncți pe baza celor de mai sus, se obține:

$$\begin{aligned} \Phi &= x_1 x_4 \cup \bar{x}_1 (x_2 x_5) \cup x_1 \bar{x}_4 (x_2 x_5) \cup \bar{x}_4 (x_1 x_3 x_5) \cup \bar{x}_1 (x_2 x_3 x_4) = \\ &= x_1 x_4 \cup \bar{x}_1 x_2 x_5 \cup x_1 x_2 \bar{x}_4 x_5 \cup x_1 x_3 \bar{x}_4 x_5 \cup \bar{x}_1 x_2 x_3 x_4 \end{aligned}$$

Acum toți termenii sunt disjuncti față de primul. Modificând mai departe termenii din dreapta ca să fie disjuncti față de al doilea, rezultă:

$$\begin{aligned}\Phi &= x_1x_4 \cup \bar{x}_1x_2x_5 \cup x_1x_2\bar{x}_4x_5 \cup x_1x_3\bar{x}_4x_5 \cup \bar{x}_5(\bar{x}_1x_2x_3x_4) = \\ &= x_1x_4 \cup \bar{x}_1x_2x_5 \cup x_1x_2\bar{x}_4x_5 \cup x_1x_3\bar{x}_4x_5 \cup \bar{x}_1x_2x_3x_4\bar{x}_5\end{aligned}$$

Repetând și pentru termenii următori aceeași procedură, relația devine:

$$\begin{aligned}\Phi(\text{disjunct}) &= x_1x_4 \cup \bar{x}_1x_2x_5 \cup x_1x_2\bar{x}_4x_5 \cup \bar{x}_2(x_1x_3\bar{x}_4x_5) \cup \bar{x}_1x_2x_3x_4\bar{x}_5 = \\ &= x_1x_4 \cup \bar{x}_1x_2x_5 \cup x_1x_2\bar{x}_4x_5 \cup x_1\bar{x}_2x_3\bar{x}_4x_5 \cup \bar{x}_1x_2x_3x_4\bar{x}_5\end{aligned}$$

Se observă că expresia obținută este complet disjunctă și nu conține termeni redundanți, deci poate fi interpretată ca o funcție de probabilitate:

$$R_{\pi} = p_1p_4 + q_1p_2p_5 + p_1p_2q_4p_5 + p_1q_2p_3q_4p_5 + q_1p_2p_3p_4q_5 \quad (3.10.4)$$

□

Dacă se pornește de la stările de succes ale sistemului, se poate folosi o metodă de minimizare derivată din cea a lui Quine și McCluskey, dar care furnizează implicații primi disjuncti [KAKAS94b]. Acest algoritm, redat în Anexa 2, are avantajul de a produce expresia de fiabilitate sub formă analitică cu numărul cel mai mic de termeni.

Fie

$$\Phi = \bigcup_{i=1}^m T_i \quad (3.10.5)$$

funcția booleană formată din termenii  $T_i$ , corespunzătorii căilor minimale (asociate funcționării sistemului). (Dacă  $T_i$  corespund nefuncționării sistemului, adică tăieturilor minimale, atunci funcția va fi  $\bar{\Phi}$ ). Această funcție se poate pune simplu sub formă de sumă disjunctă dacă se scrie sub forma:

$$\Phi = T_1 \cup T_2 \bar{T}_1 \cup T_3 \bar{T}_1 \bar{T}_2 \cup \dots \cup T_m \bar{T}_1 \bar{T}_2 \dots \bar{T}_{m-1} \quad (3.10.6)$$

În schimb problema care se pune în continuare este de a pune sub formă disjunctă și fiecare termen conținut într-un termen de forma  $T^i = T_i \bar{T}_1 \bar{T}_2 \dots \bar{T}_{i-1}$  al noii funcții. Întrucât acest termen corespunde situației ca căile  $T_j$  ( $j=1,2,\dots,i-1$ ) să fie defecte cu condiția ca  $T$  să fie bun, el se poate exprima ca:

$$\begin{aligned}T^i &= T_i (\bar{T}_1 \bar{T}_2 \dots \bar{T}_{i-1})_{\text{variabilele din } \bar{T}_{i-1}} = \\ &= T_i \overline{(T_1 \cup T_2 \cup \dots \cup T_{i-1})_{\bar{T}_{i-1}}} = T_i \text{ OP } (S_i)\end{aligned} \quad (3.10.7)$$

unde am notat  $S_i = (T_1 \cup T_2 \cup \dots \cup T_{i-1})_{\bar{T}_{i-1}}$ , adică reuniunea termenilor precedenți lui  $T_i$  în care variabilele care apar în  $T_i$  s-au înlocuit astfel încât  $\bar{T}_{i-1}=1$  (practic ei se vor șterge din acești predecesori). S-a introdus un operator OP care produce din  $S_i$  o sumă de termeni disjuncti. Astfel de operatori de disjunctizare sunt operatorul exclusiv (E-operator, exclusive operator), operatorul de combinare (CMB, combine operator), complementul

relativ (relative complement) ș.a., o listă completă a lor fiind dată în [SOH93]. În final, expresia funcției cu termeni disjunctivi va fi dată de:

$$\Phi(\text{disjunct}) = T_1 \bigcup_{i=2}^m [T_i \text{ OP } (S_i)] \quad (3.10.8)$$

**Exemplul 3.10.3.** Reiau funcția (3.10.3) din exemplul precedent pentru a explica utilizarea operatorului E. De fapt  $E(S_i)$  furnizează pe  $\bar{S}_i$  (unde  $S_i$  este o subfuncție disjunctă) sub formă de sumă logică de termeni disjunctivi:

$$\begin{aligned} S_2 &= (x_1 x_4)_{x_2=x_3=1} = x_1 x_4 \\ E(x_1 x_4) &= \bar{x}_1 \cup x_1 \bar{x}_4 \\ S_3 &= (x_1 x_4 \cup x_2 x_5)_{x_1=x_3=1} = x_4 \cup x_2 \\ E(x_4 \cup x_2) &= \bar{x}_4 \bar{x}_2 \\ S_4 &= (x_1 x_4 \cup x_2 x_5 \cup x_1 x_3 x_5)_{x_2=x_3=x_4=1} = x_1 \cup x_5 \cup x_1 = x_1 \cup x_5 \\ E(x_1 \cup x_5) &= \bar{x}_1 \bar{x}_5 \end{aligned}$$

Conform relației (3.10.8), rezultă:

$$\Phi(\text{disjunct}) = x_1 x_4 \cup x_2 x_5 (\bar{x}_1 \cup x_1 \bar{x}_4) \cup x_1 x_3 x_5 (\bar{x}_2 \bar{x}_4) \cup x_2 x_3 x_4 (\bar{x}_4 \bar{x}_5)$$

Rezultă imediat funcția de fiabilitate, echivalentă cu (3.10.4):

$$R_{st} = p_1 p_4 + p_2 p_5 (q_1 + p_1 q_4) + p_1 p_3 p_5 q_2 q_4 + p_2 p_3 p_4 q_1 q_5 \quad (3.10.9)$$

□

O altă posibilitate de calcul a lui  $T^i$  cu termeni disjunctivi este oferită de scrierea sa sub forma:

$$\begin{aligned} T^i &= (\dots((T_i \bar{T}_1) \bar{T}_2) \dots) \bar{T}_{i-1} = \\ &= (\dots((T_i \text{ OP } T_1) \text{ OP } T_2) \text{ OP } \dots) \text{ OP } T_{i-1} \end{aligned} \quad (3.10.10)$$

unde OP este un operator adecvat care produce pe  $T_i \bar{T}_j$  sub formă de termeni disjunctivi (de exemplu operatorul \$ utilizat în [VEER91]). Luând  $T^1 = T_1$ , relația de mai sus furnizează termeni disjunctivi corespunzători lui  $T^i$ , astfel încât funcția inițială sub forma unei sume de produse disjuncte va fi dată de:

$$\Phi(\text{disjunct}) = T_1 \bigcup_{i=2}^m T^i \quad (3.10.11)$$

**Exemplul 3.10.4.** Voi folosi pentru disjunctizarea funcției (3.10.3) un operator de negare NEG  $T_i$ , care furnizează pe  $\bar{T}_i$  (unde  $T_i$  este o conjuncție de variabile) sub formă de sumă booleană de termeni disjunctivi:



$$T^1 = x_1 x_4$$

$$T^2 = T_2 \text{ NEG } T_1 = x_2 x_2 (\bar{x}_1 \cup x_1 \bar{x}_4)$$

$$T^3 = (T_3 \text{ NEG } T_1) \text{ NEG } T_2 = x_1 x_3 x_5 (\bar{x}_1 \cup x_1 \bar{x}_4) (\bar{x}_2 \cup x_2 \bar{x}_5) = x_1 x_3 x_5 (\bar{x}_2 \bar{x}_4)$$

$$T^4 = ((T_4 \text{ NEG } T_1) \text{ NEG } T_2) \text{ NEG } T_3 = x_2 x_3 x_4 (\bar{x}_1 \cup x_1 \bar{x}_4) (\bar{x}_2 \cup x_2 \bar{x}_5) (\bar{x}_1 \cup x_1 \bar{x}_5 \cup x_1 \bar{x}_3 x_5) = x_2 x_3 x_4 (\bar{x}_1 \bar{x}_5)$$

Conform (3.10.11) se obține:

$$\Phi(\text{disjunct}) = x_1 x_4 \cup x_2 x_5 (\bar{x}_1 \cup x_1 \bar{x}_4) \cup x_1 x_3 x_5 (\bar{x}_2 \bar{x}_4) \cup x_2 x_3 x_4 (\bar{x}_4 \bar{x}_5)$$

care permite transpunerea directă în funcția de fiabilitate, care este aceeași cu (3.10.9).  $\square$

Torrey a propus un algoritm de disjunctizare și de calcul a fiabilității de tip terminal pornind de la setul de căi (sau tăieturi) minimale, bazat pe generarea selectivă a stărilor relevante prin alegerea și retezirea ramurilor unui arbore binar (pruned tree approach) [TORR83]. Pentru schema bloc de fiabilitate alternantă (figura 3.10.1), preluată ca referință din MIL-HDBK-217 [MIL86], funcția corespunzătoare setului de căi este:

$$\Phi = x_1 x_4 \cup x_2 x_4 \cup x_2 x_5 \cup x_3 x_5 \quad (3.10.12)$$

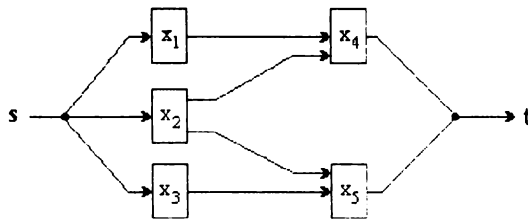


Figura 3.10.1.

Disjunctizarea se bazează pe construirea unui arbore (figura 3.10.2), alegând pentru fiecare nivel câte un element de ramificație (branching element), care primește valoarea 0 pentru ramura din stânga, respectiv 1 pentru ramura din dreapta. În mod normal arborele ar trebui să ducă la generarea celor  $2^n$  stări posibile, ceea ce ar cere un consum de timp inacceptabil de mare. Se observă însă că construcția arborelui nu mai trebuie continuată dacă se ivește una din următoarele situații:

1. A rămas o singură cale (aici  $\Phi = x_3 x_5$ ).
2. S-a găsit o cale disjunctă ( $\Phi = 1$ ).
3. Nu a mai rămas nici o cale ( $\Phi = 0$ ).

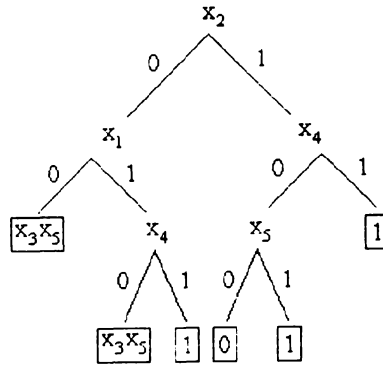


Figura 3.10.2.

Această rețezire a ramurilor care generează stări nerelevante duce la reducerea considerabilă a efortului de calcul. De pe arbore rezultă imediat funcția cu termeni disjuncți care conține toate stările favorabile:

$$\Phi(\text{disjunct}) = \bar{x}_2 \bar{x}_1 x_3 x_5 \cup \bar{x}_2 x_1 \bar{x}_4 x_3 x_5 \cup \bar{x}_2 x_1 x_4 \cup x_2 \bar{x}_4 x_5 \cup x_2 x_4 \quad (3.10.13)$$

Pentru alegerea elementului de ramificație într-un anumit nivel, se pot adopta diverse reguli de selecție:

1. Primul element al celei mai scurte căi.
2. Elementul care apare în cele mai multe căi.
3. Elementul care apare de cele mai multe ori în căile cele mai scurte.
4. Se efectuează pentru fiecare element produsul dintre lungimea celei mai scurte căi care conține acel element și unu plus numărul de căi în care nu apare acel element, iar apoi se alege elementul pentru care a ieșit valoarea minimă.

Regula 1, cea mai simplă și care găsește/elimină devreme căile lungi, duce la un timp de calcul minim, în schimb 3 și 4 necesită cele mai puține iterații.

**Simplificări în determinarea expresiei de fiabilitate.** În vederea reducerii numărului de termeni generați în suma de produse disjuncte și implicit a timpului de calcul, s-au căutat diferite criterii de ordonare a căilor/tăieturilor minimale. Tratarea lor într-o ordine aleatoare în vederea disjunctizării nu poate garanta o expresie cât mai simplă pentru funcția de fiabilitate. Aceste *tehnici de preprocesare* (preprocessing techniques), care se aplică setului de căi/tăieturi minimale înainte de a trece la modificarea termenilor astfel încât să devină disjuncți doi câte doi, pot fi în principiu următoarele [RAI78] [ABRA79][LOCKS87][WILS90]:

- Ordonarea după distanța Hamming (Hamming distance ordering). Se alege în mod arbitrar o cale/tăietură ca referință, iar celelalte se sortează în funcție de creșterea distanței Hamming față de aceasta. În acest caz termenul următor din listă va fi acela care are cele mai multe variabile comune cu termenul de referință, iar ultimul termen din listă (cel mai depărtat) va avea cele mai puține variabile comune. Termenii care au aceeași distanță Hamming pot fi luați în orice ordine între ei.

- Ordonarea lexicografică (lexicographic ordering), care constă în aranjarea termenilor corespunzători căilor/tăieturilor în ordinea crescătoare a indicilor variabilelor asociate componentelor.
- Ordonarea după cardinalitate (cardinality ordering). Se pornește de la termenul corespunzător căii/tăieturii cu cele mai puține componente și se continuă în ordinea crescătoare a numărului de variabile din fiecare termen. În caz de egalitate a dimensiunilor, termenii respectivi pot fi aranjați în orice secvență.

Soh și Rai au analizat efectul acestor criterii aplicate la 19 rețele de referință, generând fiabilitatea de tip terminal pornind de la căile minimale și nefiabilitatea pornind de la tăieturile minimale [SOH93]. S-a constatat că preprocesarea după distanța Hamming și lexicografică nu aduc avantaje la rețele mari, adică nu reduc numărul de termeni disjuncti și timpul CPU comparativ cu cele date de o ordonare aleatoare a termenilor. Ideea de bază a acestor metode este de a ține cont de variabilele comune din căi tăieturi, dar întrucât la un număr mare de componente și permutările variabilelor asociate este mare, acest avantaj dispare după primele faze de calcul. În schimb ordonarea după creșterea cardinalității reduce considerabil numărul de căi/tăieturi disjuncte și timpul CPU, fiind net superioară celorlalte metode. Acest lucru se explică prin faptul că în fazele incipiente de calcul se iau în considerare termenii cu cele mai puține variabile, care acoperă un număr mare de stări de succes și în consecință numărul de cuburi disjuncte formate față de acestea va fi mic.

Ordonarea pe baza cardinalității a căilor minimale prezintă și avantajul că termenul disjunct pentru cea mai lungă cale  $P_i$  cu  $N-1$  muchii (care trece prin toate nodurile) se obține simplu prin intersectarea termenului  $P_i$  cu complementul restului de variabile, această observație fiind exploatată de algoritmul SYREL [HARI87]. Acest algoritm este destinat pentru calculul expresiei simbolice a probabilității de comunicație ( $s, t$ ) într-un graf (orientat sau neorientat), pornind de la setul de căi dintre cele două noduri. Procedeeul de calcul evită necesitatea disjunctivării la fiecare iterație, operația cea mai mare consumatoare de timp, prin utilizarea de probabilități condiționate:

$$R = \Pr(A_1) + \Pr(A_2) \Pr(\bar{A}_1 | A_2) + \dots + \Pr(A_m) \Pr(\bar{A}_1 \cap \bar{A}_2 \cap \dots \cap \bar{A}_{m-1} | A_m) \quad (3.10.14)$$

unde  $A_i$  reprezintă evenimentul "calea  $P_i$  este operațională". În această relație trebuie calculați două feluri de termeni. Primul, bazat pe independența defectării muchiilor, este evident:

$$\Pr(A_i) = \prod_{e_k \in P_i} p_e \quad (3.10.15)$$

unde  $p_e$  reprezintă fiabilitatea muchiei  $e_e$ . Al doilea fel de termen este  $\Pr(\bar{A}_1 \cap \dots \cap \bar{A}_{i-1} | A_i)$ , care se poate scrie:

$$\Pr(\bar{A}_1 \cap \dots \cap \bar{A}_{i-1} | A_i) = \Pr(\bar{A}_{1i} \cap \dots \cap \bar{A}_{(i-1)i}) \quad (3.10.16)$$

unde  $\bar{A}_{ij}$  constituie evenimentul "calea  $P_j$  neoperațională, cu condiția că  $P_i$  este operațională". Evenimentul contrar  $A_{ij}$ , care desemnează "calea  $P_j$  operațională, cu

condiția că  $P_i$  este de asemenea operațională”, corespunde însă tocmai mulțimii elementelor care sunt în  $P_j$  dar nu sunt în  $P_i$ :

$$P_{j\bar{i}} = P_j \setminus P_i = \{e_k : e_k \in P_j \text{ și } e_k \notin P_i\} \quad (3.10.17)$$

operatorul ( $\setminus$ ) fiind numit operator exclusiv. Dar se observă că o parte din  $P_{j\bar{i}}$  se absorb, și anume aceia pentru care  $P_{k\bar{i}}$  este o submulțime a lui  $P_{j\bar{i}}$  (unde  $P_{k\bar{i}} \cap P_{j\bar{i}} = P_{k\bar{i}}$ ,  $k, l=1, 2, \dots, i-1$ ) și rămâne un set minim  $M_i$  de mulțimi condiționate corespunzătoare situației când  $P_i$  este operațională.

**Exemplul 3.10.5.** Fie căile următoare, ordonate după lungime:

$$\begin{aligned} P_1 &= \{x_1, x_3, x_5\} \\ P_1 &= \{x_2, x_5, x_6\} \\ P_1 &= \{x_2, x_4, x_6\} \\ P_1 &= \{x_1, x_3, x_4, x_6\} \end{aligned}$$

Pentru cazul când  $P_4$  este operațională se obține:

$$\begin{aligned} \Pr(\bar{A}_{1\bar{4}} \cap \bar{A}_{2\bar{4}} \cap \bar{A}_{3\bar{4}}) &= \Pr(\bar{x}_5 \cap \overline{x_2 x_5} \cap \bar{x}_2) = \Pr(\bar{x}_5 \cap (\bar{x}_2 \cup \bar{x}_5) \cap \bar{x}_2) = \\ &= \Pr(\bar{x}_5 \cap \bar{x}_2) \end{aligned}$$

Cu alte cuvinte, din seturile  $P_{1\bar{4}} = \{x_5\}$ ,  $P_{2\bar{4}} = \{x_2, x_5\}$  și  $P_{3\bar{4}} = \{x_2\}$ , al doilea a fost absorbit și a rezultat  $M_4 = \{P_{1\bar{4}}, P_{3\bar{4}}\}$ . □

Observația precedentă legată de absorbția unor termeni reduce substanțial numărul de termeni care trebuie examinați la fiecare iterație. Rezultă deci că:

$$\Pr\left(\bigcap_{j=1}^{i-1} \bar{A}_{j\bar{i}}\right) = \Pr\left(\bigcap_{P_{j\bar{i}} \in M_i} \bar{A}_{j\bar{i}}\right) \quad (3.10.18)$$

În continuare, pentru evaluarea acestei probabilități există două cazuri posibile:

a) Toate evenimentele  $\bar{A}_{j\bar{i}}$  sunt independente, adică mulțimile asociate nu conțin elemente comune luate două câte două ( $P_{j\bar{i}} \cap P_{k\bar{i}} = \emptyset$ ). În acest caz conform (3.5.8) se poate scrie simplu:

$$\Pr\left(\bigcap_{P_{j\bar{i}} \in M_i} \bar{A}_{j\bar{i}}\right) = \prod_{P_{j\bar{i}} \in M_i} \Pr(\bar{A}_{j\bar{i}}) = \prod_{P_{j\bar{i}} \in M_i} (1 - \Pr(A_{j\bar{i}})) \quad (3.10.19)$$

b) Există elemente comune printre mulțimile corespunzătoare evenimentelor  $\bar{A}_{j\bar{i}}$ . În acest caz se separă și se extrag mulțimile independente din  $M_i$ , ceea ce duce la partiționarea acesteia într-o mulțime

$$I(M_i) = \left\{ P_{j_k} : \text{pentru toate } P_{j_k} \neq P_{k_j} \in M_i, P_{j_k} \cap P_{k_j} = \emptyset \right\}$$

de seturi condiționale independente și o mulțime

$$D(M_i) = M_i - I(M_i)$$

care conține restul de seturi condiționale din  $M_i$ . Ca urmare și în formula de calcul se vor putea izola și calcula probabilitățile corespunzătoare seturilor independente:

$$\Pr \left( \bigcap_{P_{j_k} \in M_i} \bar{A}_{j_k} \right) = \Pr \left( \bigcap_{P_{k_j} \in D(M_i)} \bar{A}_{k_j} \right) \prod_{P_{j_k} \in I(M_i)} (1 - \Pr(A_{j_k})) \quad (3.10.20)$$

A mai rămas de calculat primul factor al relației (3.10.20), ceea ce se poate face descompunând cu teorema lui Bayes (dată de (3.5.12)) după fiecare element comun, sau pe baza tăieturilor asociate mulțimilor  $M_i$ . Aplicând acest ultim procedeu, se obține:

$$\Pr \left( \bigcap_{P_{k_j} \in D(M_i)} \bar{A}_{k_j} \right) = \Pr \left( \bigcup_l \bar{C}_l \right) \quad (3.10.21)$$

unde  $C_l$  reprezintă o tăietură a lui  $D(M_i)$ , adică mulțimea de elemente care, căzute toate, duc la întreruperea tuturor seturilor condiționale  $P_{k_i}$  din  $D(M_i)$ . Pentru a putea calcula probabilitatea reuniunii, fiecare tăietură  $C_l$  trebuie făcută disjunctă cu celelalte, și anume se înlocuiește cu:

$$\{\bar{e}_1, C_l\}, \{e_1, \bar{e}_2, C_l\}, \dots, \{e_1, e_2, \dots, \bar{e}_r, C_l\}$$

unde  $e_i$  reprezintă muchiile care există în  $C_k$  dar nu există în  $C_l$ . Algoritmul SYREL mai face o simplificare pentru determinarea expresiei de fiabilitate, pe baza unei observații enunțate mai înainte: dacă calea  $P_i$  are lungimea maximă  $N-1$ ,  $N$  fiind numărul de noduri, atunci probabilitatea  $\Pr(\bar{A}_1 \cap \dots \cap \bar{A}_{N-1} | A_i)$  este egală cu produsul nefiabilităților tuturor elementelor care nu apar în  $P_i$ . În concluzie, termenii disjuncți se deduc după trei proceduri separate, după următoarele categorii: (1) seturile condiționale minime nu au elemente comune, (2) există elemente comune printre aceste seturi și (3) lungimea căii este  $N-1$ . Cu aceste simplificări, complexitatea algoritmului SYREL este  $O(m^2)$  în funcție de numărul de căi  $m$ , dar exponențială în funcție de numărul de noduri  $N$ .

Buzacott a propus simplificarea formulei de fiabilitate  $(s, t)$  și globală, atât pentru grafuri neorientate cât și orientate  $G=(V, E)$ , prin ordonarea parțială a tăieturilor minime în funcție de mulțimea de noduri separată [BUZA83]. Această ordonare parțială nu este însă aplicabilă și în cazul căilor. Notând cu  $V_i$  mulțimea de noduri separate de tăietura  $i$  astfel încât nodul sursă  $s \in V_i$ , relația de ordine se definește astfel: dacă  $V_i \cap V_j$ , atunci tăieturile se află în relația  $i < j$ . Calculul fiabilității se face pe baza nefiabilității, dată de:

$$F = \Pr(C_1 \cup C_2 \cup \dots \cup C_m) = \Pr(C_1) + \Pr(\bar{C}_1 \cap C_2) + \dots + \Pr(\bar{C}_1 \cap \bar{C}_2 \cap \dots \cap \bar{C}_{m-1} \cap C_m) \quad (3.10.22)$$

unde  $C_i$  reprezintă evenimentul corespunzător tuturor muchiilor defecte din tăietura  $i$ . Dacă se notează cu  $J$  mulțimea tuturor tăieturilor cu  $k < j$  astfel încât pentru orice  $k \in J$ ,  $V_k \subset V_j$ , atunci termenii care nu fac parte din  $J$  se pot elimina, și anume:

$$\bigcap_{i=1}^{j-1} \bar{C}_i \cap C_j = \bigcap_{k \in J} \bar{C}_k \cap C_j \quad (3.10.23)$$

Astfel problema s-a redus la căutarea deconectivității în subgraful  $G_j$  corespunzător nodurilor din  $V_j$ . Este posibilă simplificarea în continuare, prin înlocuirea termenului  $C_k$  corespunzător tăieturii  $k$  a grafului  $G$ , cu termenul  $C_{kj}$  corespunzător tăieturii din  $G_j$  care separă aceeași submulțime de noduri ca și  $k$ . Practic se elimină din tăietura  $k$  arcele care nu fac parte din subgraful  $G_j$ :

$$\bigcap_{k \in J} \bar{C}_k \cap C_j = \bigcap_{k \in J} \bar{C}_{kj} \cap C_j \quad (3.10.24)$$

Avantajul acestei relații constă în faptul că astfel oricare  $C_{kj}$  va fi independent (nu conține element comun) cu  $C_j$ . Ca urmare, probabilitatea corespunzătoare se va putea scrie:

$$\Pr\left(\bigcap_{k \in J} \bar{C}_{kj} \cap C_j\right) = \Pr\left(\bigcap_{k \in J} \bar{C}_{kj}\right) \cdot \Pr(C_j) = P^s(G_j) \cdot \Pr(C_j) \quad (3.10.25)$$

unde  $P^s(G_j) = \Pr\{s \text{ conectat la toate nodurile din } G_j\}$ . Astfel calculul fiabilității în  $G$  se reduce la determinarea conectivităților în subgraurile  $G_j$  și este dată de:

$$R = 1 - F = 1 - \sum_{j=1}^m P^s(G_j) \cdot \Pr(C_j) \quad (3.10.26)$$

Revin acum la cazul când fiabilitatea de tip terminal se exprimă pornind de la funcția de structură  $\Phi$  dată de suma booleană a termenilor corespunzători căilor minime. Problema disjunctizării termenilor este îngreunată dacă numărul de variabile este mare, de aceea am propus efectuarea unei substituții în funcția de cale  $\Phi$ , bazată pe identificarea de căi de bază și căi alternative în  $G$  între nodul sursă și cel destinație [KAKAS95a]. Grupând termenii funcției în jurul căilor de bază se poate reduce numărul de termeni prin scoaterea de factor comun și în urma substituției se reduce numărul de variabile. Voi ilustra pe un exemplu.

**Exemplul 3.10.6.** Pentru rețeaua  $N_6^6$  din figura 3.10.3 funcția de cale corespunzătoare perechii de noduri  $(v_1 - v_4)$  este:

$$\Phi = x_3x_6 \cup x_4x_7 \cup x_1x_2x_6 \cup x_3x_5x_7 \cup x_4x_5x_6 \cup x_1x_2x_5x_7 \quad (3.10.27)$$

Se observă că  $\{x_1, x_2, x_6\}$  este o cale alternativă pentru calea de bază  $\{x_3, x_6\}$ , iar  $\{x_1, x_2, x_5, x_7\}$  este o alternativă pentru  $\{x_3, x_5, x_7\}$ . Grupând termenii corespunzători, se obține:

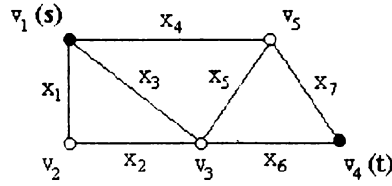


Figura 3.10.3.

$$\Phi = (x_3 \cup x_1 x_2) x_6 \cup x_4 x_7 \cup (x_3 \cup x_1 x_2) x_5 x_7 \cup x_4 x_5 x_6 \quad (3.10.28)$$

În urma substituției  $y = x_3 \cup x_1 x_2$  se obține o funcție de numai 5 variabile în loc de 7 și de 4 termeni în loc de 6:

$$\Phi = y x_6 \cup x_4 x_7 \cup y x_5 x_7 \cup x_4 x_5 x_6 \quad (3.10.29)$$

Determinarea  $\Pr(\Phi = 1)$  devine o problemă mai simplă pentru  $\Phi$  scrisă sub forma (3.10.29), decât pentru  $\Phi$  scrisă sub formă (3.10.27). Dacă se notează în ordine cu  $T_1, T_2, T_3, T_4$  termenii funcției (3.10.29), regulile de calcul probabilistic dau:

$$\begin{aligned} R_{st} &= \Pr(T_1) + \Pr(\overline{T_1} T_2) + \Pr(\overline{T_1} \overline{T_2} T_3) + \Pr(\overline{T_1} \overline{T_2} \overline{T_3} T_4) = \\ &= \Pr(y x_6) + \Pr(y x_6 x_4 x_7) + \Pr(\overline{x_6} \overline{x_4} y x_5 x_7) + \Pr(\overline{y_6} \overline{x_7} x_4 x_5 x_6) = \\ &= p_y p_6 + (1 - p_y p_6) p_4 p_7 + q_6 p_4 p_y p_5 p_7 + q_y q_7 p_4 p_5 p_6 \end{aligned} \quad (3.10.30)$$

Singura problemă a mai rămas calcularea lui  $p_y$ , funcția de substituție fiind însă simplă, disjunctizarea ei este imediată:

$$\begin{aligned} p_y &= \Pr(y = 1) = \Pr(x_3 \cup x_1 x_2) = \Pr(x_3 \cup x_1 x_2 \overline{x_3}) = \\ &= \Pr(x_3) + \Pr(x_1 x_2 \overline{x_3}) = p_3 + p_1 p_2 q_3 \end{aligned} \quad (3.10.31)$$

Înlocuind (3.10.31) în (3.10.30) se obține ușor expresia finală a fiabilității de tip terminal:

$$\begin{aligned} R_{st} &= (p_3 + p_1 p_2 q_3) p_6 + p_4 p_7 - (p_3 + p_1 p_2 q_3) p_4 p_6 p_7 + (p_3 + \\ &+ p_1 p_2 q_3) q_4 p_5 q_6 p_7 + p_4 p_5 p_6 q_7 - (p_3 + p_1 p_2 q_3) p_4 p_5 p_6 q_7 \end{aligned} \quad (3.10.32)$$

□

Calculul fiabilității pe baza sumei disjuncte (3.10.6) impune la pasul  $i$ , așa cum am arătat, calculul unui termen de forma  $T_1 \cup T_2 \cup \dots \cup T_{i-1}$ , în care se elimină variabilele conținute în  $T_i$ . (Se constată aici o justificare a necesității de a ordona termenii după creșterea cardinalității, în speranța de obținere a unor expresii  $T_1 \cup T_2 \cup \dots \cup T_{i-1}$  cât mai simple, cu cât mai puține variabile). După eliminarea variabilelor în această sumă booleană pot rămâne și termeni redundanți, care dispar pe baza absorbției sau idempotenței. Pentru eliminarea timpurie a acestora am pus la punct o metodă sistematică, pe care o prezint sub forma unui exemplu.

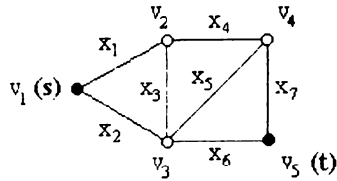


Figura 3.10.4.

**Exemplul 3.10.7.** Se consideră rețeaua din figura 3.10.4 în care se cere fiabilitatea comunicației  $(s, t)$ . Termenii  $T_i$  corespunzători căilor minime sunt trecuți în tabelul 3.10.5 în ordinea crescătoare a numărului de variabile. Pentru fiecare astfel de termen se formează o mulțime de noi termeni  $J$ , obținuți din cei precedenți cu eliminarea variabilelor conținute în  $T_i$  și excluderea acelor termeni  $T_k \in J$  pentru care există  $T_j \in J$  astfel încât  $\{T_j\} \subseteq \{T_k\}$ ,  $j \neq k$ , unde  $\{T_j\}$  exprimă mulțimea variabilelor din termenul  $T_j$ . Termenii  $T_i$  și  $T_j$  rămași în tabelul 3.10.5 permit exprimarea ușoară a expresiei analitice de fiabilitate, și anume:

$$\begin{aligned}
 R_{st} &= \sum_{i=1}^7 \Pr \left\{ \bigcap_{j=1}^{i-1} \bar{T}_j T_i \right\} = \sum_{i=1}^7 \Pr \left\{ \bigcap_{k \in J} \bar{T}_k T_i \right\} = \\
 &= \Pr(x_2 x_6) + \Pr(x_2 x_6 \cap x_1 x_4 x_7) + \Pr(\bar{x}_2 \cap x_4 x_7 \cap x_1 x_3 x_6) + \\
 &+ \Pr(\bar{x}_6 \cap x_1 x_4 \cap x_2 x_5 x_7) + \Pr(\bar{x}_2 \cap \bar{x}_7 \cap \bar{x}_3 \cap x_1 x_4 x_5 x_6) + \\
 &+ \Pr(\bar{x}_4 \cap \bar{x}_6 \cap \bar{x}_2 \cap x_1 x_3 x_5 x_7) + \Pr(\bar{x}_6 \cap \bar{x}_1 \cap \bar{x}_5 \cap x_2 x_3 x_4 x_7) = \\
 &= p_2 p_6 + (1 - p_2 p_6) p_1 p_4 p_7 + (1 - p_2)(1 - p_4 p_7) p_1 p_3 p_6 + \\
 &+ (1 - p_6)(1 - p_1 p_4) p_2 p_5 p_7 + (1 - p_2)(1 - p_7)(1 - p_3) p_1 p_4 p_5 p_6 + \quad (3.10.33) \\
 &+ (1 - p_4)(1 - p_6)(1 - p_2) p_1 p_3 p_5 p_7 + (1 - p_6)(1 - p_1)(1 - p_5) p_2 p_3 p_4 p_7
 \end{aligned}$$

Ceea ce mai este foarte important de remarcat, că în cadrul fiecărui termen factorii produsului logic au variabile diferite, deci corespund la evenimente independente și pentru calculul probabilității corespunzătoare se poate aplica direct formula (3.5.8).

$i$	$T_i$	$J$
1	$x_2 x_6$	—
2	$x_1 x_4 x_7$	$x_2 x_6$
3	$x_1 x_3 x_6$	$x_2, x_4 x_7$
4	$x_2 x_5 x_7$	$x_6, x_1 x_4, \bar{x}_1 x_3 x_6$
5	$x_1 x_4 x_5 x_6$	$x_2, x_7, x_3, \bar{x}_2 x_6$
6	$x_1 x_3 x_5 x_7$	$\bar{x}_2 x_6, x_4, x_6, x_2, \bar{x}_4 x_6$
7	$x_2 x_3 x_4 x_7$	$x_6, x_1, \bar{x}_1 x_6, x_5, \bar{x}_1 x_5 x_6, \bar{x}_1 x_5 x_7$

Tabelul 3.10.5.

○ altă posibilitate de simplificare a calculului de fiabilitate se obține prin descompunerea funcției de structură  $\Phi$  în subfuncții disjuncte mai puțin complexe și cu



un număr mai mic de variabile. Metoda este utilă dacă numărul de componente este mare, iar noile funcții rezultate se pot trata comod pentru disjunctizare pe câte o diagramă Karnaugh, în ideea că descompunerea se face astfel încât aceste funcții să aibă cel mult 6 variabile. Descompunerea funcției  $\Phi$  după anumite variabile poate urma două moduri de abordare:

- Descompunerea după *strategia exhaustivă* (exhaustive policy), care se face pentru toate cele  $2^k$  combinații posibile al variabilelor selectate  $x_1, x_2, \dots, x_k$ . Relația de descompunere este:

$$\begin{aligned} \Phi &= x_1 x_2 \dots x_k \Phi_{x_1=1, x_2=1, \dots, x_k=1} \cup \bar{x}_1 x_2 \dots x_k \Phi_{x_1=0, x_2=1, \dots, x_k=1} \cup \dots \cup \bar{x}_1 \bar{x}_2 \dots \bar{x}_k \Phi_{x_1=0, x_2=0, \dots, x_k=0} = \\ &= x_1 x_2 \dots x_k \Phi_1 \cup \bar{x}_1 x_2 \dots x_k \Phi_2 \cup \dots \cup \bar{x}_1 \bar{x}_2 \dots \bar{x}_k \Phi_{2^k} \end{aligned} \quad (3.10.34)$$

De fapt metoda restrânge tratarea funcției originale pe subdomenii, conform diagramei din figura 3.10.6.a (unde s-a luat  $k=3$ ).

- Descompunerea după *strategia conservativă* (conservative policy) se bazează pe înjumătățirea succesivă a domeniului funcției (figura 3.10.6.b). Dezvoltarea va conduce în acest caz la numai  $k+1$  funcții parțiale, dar acestea nu vor mai fi echilibrate ca și complexitate ( $\Phi_1$  este mai complexă decât  $\Phi_2$ , aceasta decât  $\Phi_3$ , ș.a.m.d.):

$$\begin{aligned} \Phi &= x_1 \Phi_{x_1=1} \cup \bar{x}_1 x_2 \Phi_{x_1=0, x_2=1} \cup \dots \cup \bar{x}_1 \bar{x}_2 \dots \bar{x}_k \Phi_{x_1=0, x_2=0, \dots, x_k=0} = \\ &= x_1 \Phi_1 \cup \bar{x}_1 x_2 \Phi_2 \cup \dots \cup \bar{x}_1 \bar{x}_2 \dots \bar{x}_k \Phi_{k+1} \end{aligned} \quad (3.10.35)$$

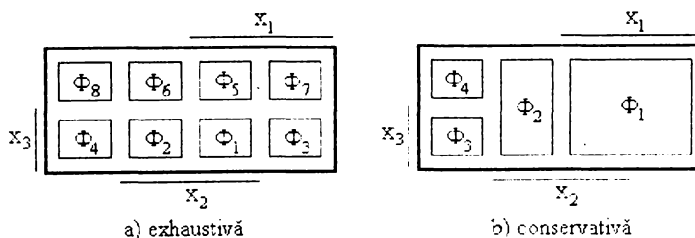


Figura 3.10.6.

Este clar și din figura 3.10.6 că termenii dezvoltării vor fi disjuncți, deci problema se reduce la disjunctizarea noilor funcții  $\Phi_i$ . Acestea permit o tratare mai ușoară, deoarece conțin un număr mai mic de variabile. Pentru a obține expresii cât mai simple pentru funcțiile  $\Phi_i$ , este de dorit ca variabilele după care se face descompunerea, și care vor fi eliminate, să fie cele care apar în cei mai mulți termeni din funcția de structură  $\Phi$ . Întrucât pentru fiabilitatea de tip terminal aceasta se exprimă pe baza căilor/tăieturilor minimale, se pot alege pentru descompunere variabilele corespunzătoare muchiilor incidente la nodul sursă și/sau destinație, care apar de regulă în cele mai multe căi/tăieturi. Strategia conservativă are avantajul de a produce mai puține funcții parțiale, dar care diferă ca și

complexitate, pe când la strategia exhaustivă toate funcțiile sunt echilibrate, dar numărul lor este mai mare.

**Exemplul 3.10.8.** Voi prezenta calculul fiabilității prin factorizarea disjunctă a funcției  $\Phi$  dată de (3.10.27), după o strategie exhaustivă pentru rețeaua din figura 3.10.3. Alegând pentru descompunere variabilele asociate nodului destinație, se obține:

$$\Phi = x_6 x_7 \Phi_{\substack{x_6=1 \\ x_7=1}} \cup \bar{x}_6 x_7 \Phi_{\substack{x_6=0 \\ x_7=1}} \cup x_6 \bar{x}_7 \Phi_{\substack{x_6=1 \\ x_7=0}} \cup \bar{x}_6 \bar{x}_7 \Phi_{\substack{x_6=0 \\ x_7=0}} \quad (3.10.36)$$

Ultimul termen fiind evident zero ( $x_6=0$  și  $x_7=0$  întrerup orice cale) și aplicând teorema absorbției, rezultă:

$$\Phi = x_6 x_7 (x_3 \cup x_4 \cup x_1 x_2) \cup \bar{x}_6 x_7 (x_4 \cup x_3 x_5 \cup x_1 x_2 x_5) \cup x_6 \bar{x}_7 (x_3 \cup x_1 x_2 \cup x_4 x_5) \quad (3.10.37)$$

Cei trei termeni ai funcției (3.10.37) fiind reciproc exclusivi, problema se reduce la disjunctizarea următoarelor trei funcții simple:

$$\Phi_1 = x_3 \cup x_4 \cup x_1 x_2 = x_3 \cup \bar{x}_3 x_4 \cup \bar{x}_3 \bar{x}_4 x_1 x_2 \quad (3.10.38.a)$$

$$\Phi_2 = x_4 \cup x_3 x_5 \cup x_1 x_2 x_5 = x_4 \cup \bar{x}_4 x_3 x_5 \cup \bar{x}_4 \bar{x}_3 x_1 x_2 x_5 \quad (3.10.38.b)$$

$$\Phi_3 = x_3 \cup x_1 x_2 \cup x_4 x_5 = x_3 \cup \bar{x}_3 x_1 x_2 \cup \bar{x}_3 \bar{x}_1 x_4 x_5 \cup \bar{x}_3 x_1 \bar{x}_2 x_4 x_5 \quad (3.10.38.c)$$

Pe baza relațiilor (3.10.37) și (3.10.38) rezultă imediat funcția de fiabilitate căutată:

$$R_{..} = p_6 p_7 (p_3 + q_3 p_4 + p_1 p_2 q_3 q_4) + q_6 p_7 (p_4 + p_3 q_4 p_5 + p_1 p_2 q_3 q_4 p_5) + p_6 q_7 (p_3 + p_1 p_2 q_3 + q_1 q_3 p_4 p_5 + p_1 q_2 q_3 p_4 p_5) \quad (3.10.39)$$

Această expresie simbolică a fiabilității este desigur echivalentă cu (3.10.32), ambele fiind obținute pe baza setului de căi minimale, dar după procedee diferite.  $\square$

În anumite situații particulare obținerea funcției de fiabilitate este ușurată de proprietățile pe care le posedă graful asociat rețelei. Astfel, pentru fiabilitatea globală (sursă la toate terminalele) SAT (source-to-all terminal) într-o rețea orientată este adevărată următoarea teoremă [HIGA94]:

**Teorema 3.10.9.** Condiția necesară și suficientă ca să existe un arbore de acoperire în graful orientat  $G = (V, E)$  referitor la SAT, este ca să existe cel puțin câte un arc incident către fiecare nod din  $V - \{s\}$ , unde  $s$  reprezintă nodul sursă.

*Demonstrație.* Presupunând că există un arbore de acoperire al lui  $G$ , va exista câte o cale de la  $s$  la toate celelalte noduri, deci trebuie să existe câte un arc incident către fiecare nod. Invers, presupunând că există câte un arc incident către fiecare nod diferit de  $s$ , pornind de la nodul sursă se poate construi din aproape în aproape un arbore orientat care să cuprindă toate nodurile grafului.  $\square$

Bazat pe această teoremă, funcția booleană de structură pentru calculul fiabilității SAT se poate exprima ușor sub formă de produse de factori, astfel încât fiecare factor corespunde unui nod  $v \neq s$  al grafului:

$$\Phi(G) = \prod_{v \in V - \{s\}} \bigcup_{e_i \in E_i(v)} x_i \quad (3.10.40)$$

Cu  $E_i(v)$  am notat mulțimea arcelor incidente către nodul  $v$ . Fiecare factor al acestei funcții corespunde la evenimente independente, transpunerea în relație de probabilități necesită deci numai disjunctizarea termenilor din fiecare factor, ceea ce decurge simplu datorită numărului redus de variabile (cele mai multe variabile sunt conținute în factorul corespunzător nodului cu gradul de incidență maxim).

**Exemplul 3.10.10.** Rețeaua orientată de tip punte din figura 3.10.7.a satisface condiția cerută de teorema 3.10.9. Funcția de structură care exprimă condiția ca să existe un arc valid către fiecare nod diferit de cel sursă, este:

$$\Phi(G) = x_1(x_2 \cup x_3)(x_4 \cup x_5) \quad (3.10.41)$$

Pentru ca expresia să fie transpusă în  $R(G)$ , trebuie disjunctizați termenii din paranteze:

$$\Phi(\text{disjunct}) = x_1(x_2 \cup \bar{x}_2 x_3)(x_4 \cup \bar{x}_4 x_5) \quad (3.10.42)$$

ceea ce duce la:

$$R(G) = p_1(p_2 + q_2 p_3)(p_4 + q_4 p_5) \quad (3.10.43)$$

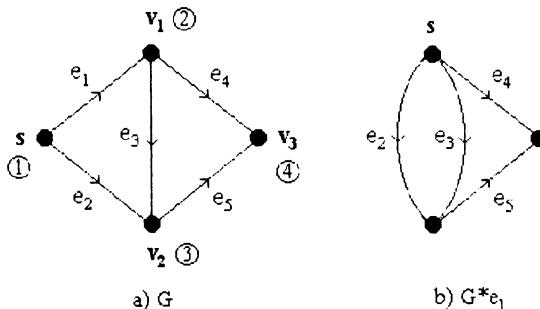


Figura 3.10.7.

Pentru obținerea funcției de structură corespunzătoare fiabilității SAT sub forma (3.10.40), Higashiyama et al. au propus două metode bazate pe descompunerea grafului [HIGA94]. Metoda formulei factorizate (factored formula method, F method) se bazează pe aplicarea formulei pivotale de factorizare prin unificarea nodurilor într-o ordine topologică. Nodurile se numerotează astfel încât dacă un arc merge de la  $v_i$  la  $v_j$ , atunci  $i < j$  (figura 3.10.7.a). Tehnica de reducere se aplică până când rămâne un singur nod, numit super nod sursă. Pentru cazul din exemplul 3.10.10, factorizarea după  $e_1$  dă:

$$\Phi(G) = x_1 \Phi(G^* e_1) \cup \bar{x}_1 \Phi(G - e_1) \quad (3.10.44)$$

unde cu  $G^* e_1$  am notat graful obținut din  $G$  prin contractarea muchiei  $e_1$  (figura 3.10.7.b), iar cu  $G - e_1$  graful obținut prin ștergerea lui  $e_1$ . Se observă însă că al doilea termen din (3.10.44) este zero, deoarece în  $G - e_1$  nu există arc incident către  $v_1$ . Urmează mai departe:

$$\Phi(G^* e_1) = (x_2 \cup x_3) \Phi(G^*(e_1, e_2 | e_3)) \cup \overline{(x_2 \cup x_3)} \Phi(G - (e_1, e_2 | e_3)) \quad (3.10.45)$$

Se observă că datorită ordinii de numerotare se obține o reducere a numărului de termeni, în fiecare dezvoltare al doilea termen fiind nul. Ordonarea nodurilor este posibilă însă numai dacă graful este lipsit de cicluri. Pentru un graf  $G$  cu cicluri, acesta se înlocuiește cu un graf condensat  $G^*$  (aciclic) astfel încât fiecare componentă tare conexă a lui  $G$  se înlocuiește cu câte un singur nod. Metoda termenului singular de produse (single product term method, S method) se bazează pe descompunerea grafului (ciclic) în grafuri secțiune (vertex-section graphs)  $G_k$ , astfel încât fiecare să conțină câte o componentă tare conexă a lui  $G$ . Astfel pentru  $\Phi(G)$  rezultă o formulă sub formă de produs boolean:

$$\Phi(G) = \prod_{k=1}^{|G^*|} \Phi(G_k) \quad (3.10.46)$$

unde  $|G^*|$  reprezintă numărul de noduri ale grafului condensat  $G^*$ . Factorii  $\Phi(G_k)$  se calculează mai departe aplicând metoda formulei factorizate pentru fiecare graf  $G_k$ .

În continuare, ca o contribuție personală, voi prezenta o simplificare a determinării expresiei de fiabilitate bazată pe *decompoziția funcției* booleene, metodă adaptată de la sinteza circuitelor logice combinaționale. Ideea de bază o constituie faptul că o dezvoltare exhaustivă a funcției de structură  $\Phi$  după formula (3.10.34) creează  $2^k$  funcții parțiale noi, este adevărat mai simple, care trebuie disjunctizate. Ar fi de dorit însă ca descompunerea să creeze cât mai puține funcții noi, de preferință una singură. Acest lucru se poate obține prin *decompoziția disjunctă* a funcției de structură după un număr de variabile libere  $\{x_1, x_2, \dots, x_k\}$ , adică:

$$\Phi(x_1, \dots, x_n) = \bigvee [x_1, \dots, x_k, f_1(x_{k+1}, \dots, x_n)] \quad (3.10.47)$$

Schematic *decompoziția disjunctă* (care partiționează setul de variabile în submulțimi disjuncte) este ilustrată în figura 3.10.8, pe care se observă că în loc de tratarea unei funcții cu  $n$  variabile este necesară tratarea unei funcții cu  $k+1$  variabile și a unei subfuncții  $f_1$  cu  $n-k$  variabile. Problemele care se pun sunt însă următoarele:

- (I) dacă funcția  $\Phi$  este decompozabilă
- (II) dacă da, atunci funcția  $\Phi$  dată de (3.10.47) să fie scrisă explicit sub forma unei sume booleene de termeni disjuncți (în sensul de a corespunde la evenimente mutual exclusive), pentru a se putea transpune în relație de probabilitate

Pentru a răspunde la prima întrebare, trebuie întocmite tabelele de decompoziție (în număr de  $2^{n-1}-1$ ), pe care se încearcă identificarea de variabile libere și se caută verificarea condiției de decompozabilitate. Tabelele de decompoziție sunt diagrame Karnaugh încadrate conform tuturor distribuțiilor posibile în două submulțimi disjuncte a variabilelor funcției. Funcția  $\Phi(x_1, \dots, x_n)$  admite o decompoziție simplă disjunctă de forma (3.10.47) dacă și numai dacă frecvența coloanelor în tabela de decompoziție  $\{\{x_1, \dots, x_k\}, \{x_{k+1}, \dots, x_n\}\}$  este  $v \leq 2$  [SZIT90]. În acest caz  $\{x_1, \dots, x_k\}$  devin variabile libere, iar subfuncția  $f_1$  se va asocia unuia din rândurile tabeli de decompoziție. (Întrucât prin rotirea cu  $90^\circ$  coloanele se interschimbă cu liniile, condiția de mai înainte trebuie verificată și pentru linii). Dacă funcția este decompozabilă, atunci însăși metoda decompoziției duce la o formă disjunctivă cu termeni disjuncti, în concordanță cu problema (II), așa cum se va vedea pe exemplul care urmează. Deci rămâne de disjunctizat doar funcția  $f_1$  cu număr redus de variabile, ceea ce se face prin oricare din metodele de la începutul acestui paragraf.

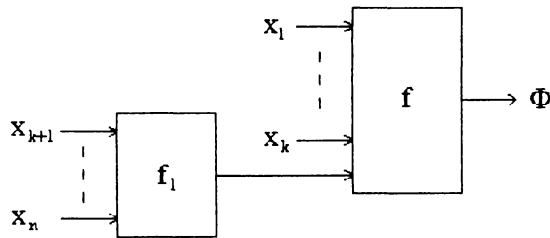


Figura 3.10.8.

**Exemplul 3.10.12.** Funcția de structură corespunzătoare căilor minimale ( $s, t$ ) pentru rețeaua din figura 3.10.9 este:

$$\Phi = x_1x_2 \cup x_1x_3 \cup x_4 \quad (3.10.48)$$

Tabelele de decompoziție pentru această funcție sunt date în figura 3.10.10. Se observă că condiția de a avea numai două tipuri de coloane este satisfăcută pentru cazurile d) și g), însemnate cu \*. Voi ilustra decompoziția numai pentru cazul g), în care variabilele libere sunt  $x_1$  și  $x_4$ . Subfuncția  $f_1$  se poate exprima din rândul lui  $x_1\bar{x}_4$ , cel mai ușor sub formă negată

$$\bar{f}_1 = \bar{x}_2\bar{x}_3$$

adică

$$f_1 = x_2 \cup x_3$$

care sub formă disjunctă se scrie simplu conform (3.10.2) ca:

$$f_1 = x_2 \cup \bar{x}_2x_3 \quad (3.10.49)$$

Exprimând acum  $\Phi$  după rândurile tabeli 3.10.10.g, se obține decompoziția acesteia:

$$\begin{aligned}\Phi &= \bar{x}_1\bar{x}_4(0) \cup \bar{x}_1x_4(1) \cup x_1x_4(1) \cup x_1\bar{x}_4f_1 = \\ &= x_4 \cup x_1\bar{x}_4(x_2 \cup \bar{x}_2x_3)\end{aligned}\quad (3.10.50)$$

Această expresie conține numai termeni disjunctivi și se poate transpune direct într-o relație de probabilitate, obținându-se fiabilitatea comunicației  $(s, t)$ :

$$R_{st} = p_4 + p_1q_4(p_2 + q_2p_3) \quad (3.10.51)$$

□

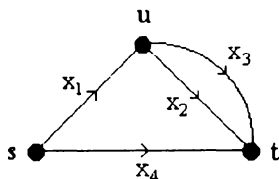


Figura 3.10.9.

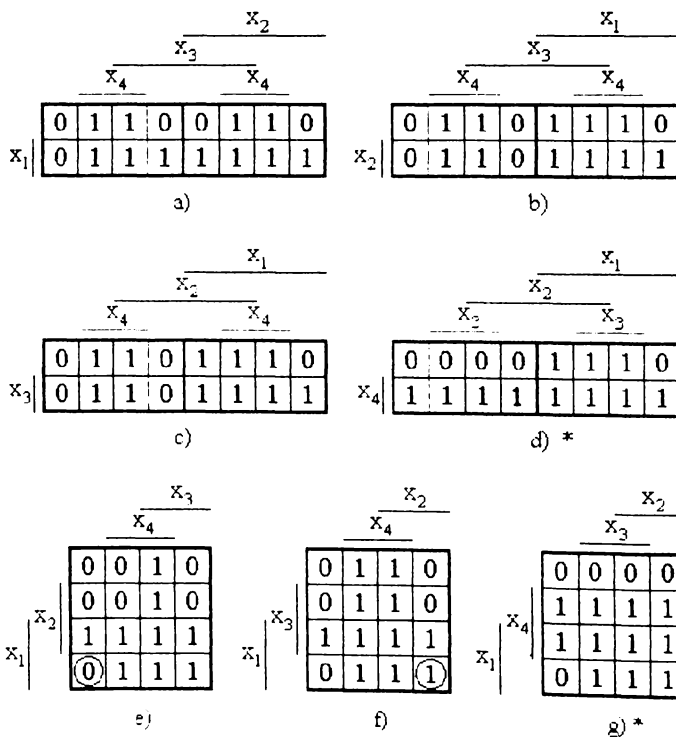


Figura 3.10.10.

Uneori decompoziția este posibilă numai prin adăugarea sau eliminarea de termeni minimali. Făcând apoi corecția cuvenită asupra funcției modificate  $\Phi_M$ , se poate ajunge ușor la o formă care să permită transpunerea în relație de probabilitate. Astfel, pentru cazul din figura 3.10.10.e, prin adăugarea termenului încercuit  $x_1\bar{x}_2\bar{x}_3\bar{x}_4$ , funcția devine decompozabilă ( $f_1 = x_4$ ), și anume:

$$\begin{aligned}\Phi_M &= \bar{x}_1\bar{x}_2f_1 \cup \bar{x}_1x_2f_1 \cup x_1x_2(1) \cup x_1\bar{x}_2(1) = \\ &= \bar{x}_1f_1 \cup x_1 = \bar{x}_1x_4 \cup x_1\end{aligned}\quad (3.10.52)$$

Aplicând în final corecția, rezultă

$$\Phi = \Phi_M \overline{x_1\bar{x}_2\bar{x}_3\bar{x}_4} \quad (3.10.53)$$

sau dacă se exprimă negata:

$$\bar{\Phi} = \bar{\Phi}_M \cup x_1\bar{x}_2\bar{x}_3\bar{x}_4 \quad (3.10.54)$$

Se observă pe diagramă că termenul adăugat este disjunct față de  $\bar{\Phi}_M$ , și se știe că  $\Phi_M$  exprimat conform (3.10.52) se poate transpune în relație de probabilitate. Fiabilitatea căutată se va putea calcula din  $\bar{\Phi}$ , ca fiind:

$$\begin{aligned}R_{st} &= \Pr\{\Phi = 1\} = 1 - \Pr\{\bar{\Phi} = 1\} = 1 - \Pr\{\bar{\Phi}_M \cup x_1\bar{x}_2\bar{x}_3\bar{x}_4 = 1\} = \\ &= 1 - \Pr\{\bar{\Phi}_M = 1\} - \Pr\{x_1\bar{x}_2\bar{x}_3\bar{x}_4\} + \Pr\{\Phi_M = 1\} - \Pr\{x_1\bar{x}_2\bar{x}_3\bar{x}_4\} = \\ &= \Pr\{\bar{x}_1x_4 \cup x_1\} - \Pr\{x_1\bar{x}_2\bar{x}_3\bar{x}_4\} = q_1p_4 + p_1 - p_1q_2q_3q_4\end{aligned}\quad (3.10.55)$$

Evident expresia (3.10.55) este echivalentă cu (3.10.51).

Dacă se recurge la eliminarea de mintermi pentru a face posibilă decompoziția, calculele decurg în mod similar, doar relația de corecție este o sumă booleană în loc de produs boolean. De exemplu, în tabelul din figura 3.10.10.e, decompoziția devine posibilă prin eliminarea termenului încercuit  $x_1x_2\bar{x}_3\bar{x}_4$ . Subfuncția este  $f_1 = x_4$ , iar decompoziția funcției modificate duce la:

$$\begin{aligned}\Phi_M &= \bar{x}_1\bar{x}_3f_1 \cup \bar{x}_1x_3f_1 \cup x_1x_3(1) \cup x_1\bar{x}_3f_1 = \\ &= \bar{x}_1f_1 \cup x_1x_3 \cup x_1\bar{x}_3f_1 = \\ &= \bar{x}_1x_4 \cup x_1x_3 \cup x_1\bar{x}_3x_4\end{aligned}\quad (3.10.56)$$

Făcând corecția, se obține:

$$\Phi = \Phi_M \setminus x_1x_2\bar{x}_3\bar{x}_4 \quad (3.10.57)$$

Termenii funcției (3.10.57) fiind evident disjuncti, se poate scrie direct expresia pentru fiabilitatea conectivității  $(s, t)$ :

$$\begin{aligned}
 R_{\bar{x}} &= \Pr\{\Phi = 1\} = \Pr\{\overline{\Phi}_M = 1\} + \Pr\{x_1 x_2 \bar{x}_3 \bar{x}_4\} = \\
 &= \Pr\{\bar{x}_1 x_4 \cup x_1 x_3 \cup x_1 \bar{x}_3 x_4\} + \Pr\{x_1 x_2 \bar{x}_3 \bar{x}_4\} = \\
 &= q_1 p_4 + p_1 p_3 + p_1 q_3 p_4 + p_1 p_2 q_3 q_4
 \end{aligned} \tag{3.10.58}$$

Bineînțeles că și relația (3.10.58) este echivalentă cu (3.10.51) și (3.10.55).

În concluzie, utilizarea decompoziției funcției de structură booleene reduce efortul necesar pentru disjunctizarea termenilor la disjunctizarea unor subfuncții simple cu număr redus de variabile. Se poate căuta și o partiționare a mulțimii variabilelor în mai mult de două submulțimi, care să conducă la o decompoziție compusă. Efortul de calcul este orientat cu precădere spre întocmirea și verificarea tabelelor de decompoziție, în număr relativ mare pentru un număr mare de variabile, dar pentru acestea există programe de decompoziție pe calculator [ARATÓ90]. Avantajele aplicării decompoziției la determinarea funcției de fiabilitate rezidă în faptul că se obține ușor o formă simplă a expresiei analitice de fiabilitate și pentru aceasta se apelează la proceduri de decompoziție existente, puse la punct și utilizate pentru alte scopuri (la proiectarea schemelor logice combinaționale).

### 3.11. Evaluarea fiabilității unei rețele cu relații logice de semnal

O posibilitate pentru evaluarea fiabilității aferente comunicației ( $s, t$ ) constă în utilizarea relațiilor logice de semnal (logical signal relations) [AGGA78]. Problema de a găsi probabilitatea existenței unei conexiuni între nodul sursă și cel terminal se transpune aici în probabilitatea ca un semnal emis în nodul sursă să ajungă la nodul destinație:

$$R_{\tau} = \Pr\{s \text{ conectat cu } t\} = \Pr\{\text{semnalul din } s \text{ ajunge la } t\} \tag{3.11.1}$$

Avantajul principal al metodei este că nu necesită o cunoaștere prealabilă a nici unei căi sau tăieturi din graf, care sunt condiții de bază la multe alte metode, așa cum s-a prezentat la paragraful 3.8.

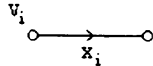
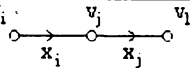
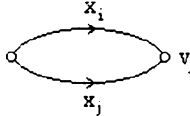
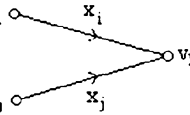
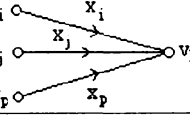
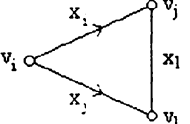
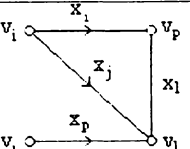
Relațiile logice de semnal se pot utiliza pentru găsirea expresiei simbolice de fiabilitate atât în cazul unui graf orientat, cât și neorientat. Dacă este vorba însă de un graf neorientat, acesta trebuie transformat prin fixarea direcțiilor unor muchii, astfel încât muchiile incidente la nodul sursă să aibă originea în aceasta, iar cele incidente la nodul destinație să fie orientate către acest nod. Procedura impune automat și direcțiile altor muchii, cele neafectate rămânând neorientate. La stabilirea sensului muchiilor se are în vedere ca rețeaua să nu conțină cicluri orientate sau bucle. Numerotarea nodurilor și muchiilor, pornind de la nodul sursă se face pe baza convenției de numerotare TNC (Terminal Numbering Convention): nodului muchiei următoare fiecărui nod muchie, în sensul săgeților, i se atribuie un număr mai mare decât acesteia, având grijă ca să nu existe două noduri/muchi cu același număr.

Pentru subgrafurile uzuale, relațiile logice de semnal folosite la calculul fiabilității de tip ( $s, t$ ), presupunând noduri perfecte, sunt date în tabelul 3.11.1. Fiecare relație este astfel scrisă încât să conțină termeni reciproc exclusivi, ceea ce permite transpunerea directă a relațiilor în expresii de fiabilitate. Notațiile noi folosite sunt:

$S(v_i)$  - prezența logică a semnalului la nodul  $v_i$

$\overline{v_i v_j}$  - fuziunea nodurilor  $v_i$  și  $v_j$



Nr.	Subgraful	Relația logică de semnal
1.		$S(v_j) = x_i S(v_i)$ $S(\overline{v_i v_j}) = S(v_i)$
2.		$S(v_l) = x_i S(v_j) = x_i x_j S(v_i)$
3.		$S(v_j) = (x_i \cup \bar{x}_i x_j) S(v_i)$
4.		$S(v_l) = x_i \bar{x}_j S(v_i) \cup \bar{x}_i x_j S(v_i) \cup x_i x_j S(\overline{v_i v_j})$
5.		$S(v_l) = [x_i \bar{x}_j S(v_i) \cup \bar{x}_i x_j S(v_i) \cup x_i x_j S(\overline{v_i v_j})] \cup \bar{x}_i \bar{x}_j x_p S(v_p)$
6.		$S(v_j) = x_i S(v_i) \cup \bar{x}_i x_j x_l S(v_i)$ $S(v_l) = x_j S(v_i) \cup x_i \bar{x}_j x_l S(v_i)$ $S(\overline{v_j v_l}) = x_i S(v_i) \cup \bar{x}_i \bar{x}_j S(v_i)$
7.		$S(\overline{v_p v_l}) = (x_i \bar{x}_j \bar{x}_p \cup x_i \bar{x}_j x_p) S(v_i) \cup \bar{x}_i \bar{x}_j x_p S(v_j) \cup (x_i \bar{x}_j x_p \cup x_j x_p) S(\overline{v_j v_l})$

Tabelul 3.11.1.

Cu aceste notații evident:

$$S(\overline{v_i v_j}) = S(v_i) \cup S(v_j) \quad (3.11.2)$$

Menționez că în [AGGA78] pentru subrețeaua nr.7 este dată o altă formulă, și anume:

$$S(\overline{v_j v_l}) = (x_i \bar{x}_j \cup x_j \bar{x}_j) S(v_i) \cup x_p \bar{x}_j S(v_j) \cup x_i x_j x_p S(\overline{v_i v_j}) \quad (3.11.3)$$

Voi demonstra însă că această relație este greșită, și cea dată de mine este corectă, deoarece relația (3.11.3) nu acoperă toate situațiile posibile pentru ca semnalul să ajungă în  $v_p$  sau  $v_l$ . Această condiție evident se poate scrie:

$$S(\overline{v_p v_i}) = (x_i \cup x_j) S(v_i) \cup x_p S(v_j) \quad (3.11.4)$$

Întrucât (3.11.4) este o funcție de variabile binare, se poate transpune în diagrama Karnaugh din figura 3.11.1. Dacă pe aceasta se marchează termenii corespunzători relației (3.11.3), se constată că rămân două cazuri favorabile neacoperite:  $x_i \bar{x}_j x_p \overline{S(v_i) S(v_j)}$  și  $\bar{x}_i x_j x_p S(v_i) \overline{S(v_j)}$ . Pentru corecție propun gruparea disjunctă de termeni din figura 3.11.2, pe baza căreia se poate scrie relația lipsită de eroare:

$$S(\overline{v_p v_i}) = (x_i \bar{x}_j \bar{x}_p \cup x_j \bar{x}_p) S(v_i) \cup \bar{x}_i \bar{x}_j x_p S(v_j) \cup (x_i \bar{x}_j x_p \cup x_j x_p) S(\overline{v_i v_j}) \quad (3.11.5)$$

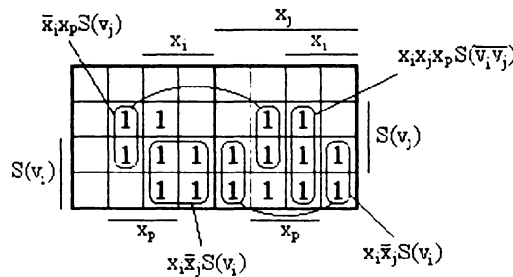


Figura 3.11.1.

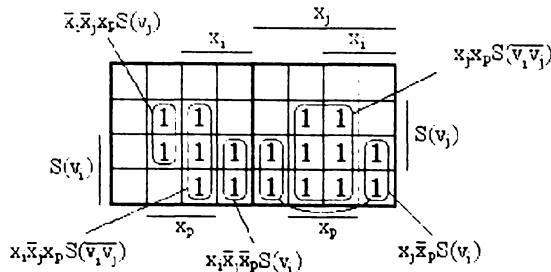


Figura 3.11.2.

Algoritmul de determinare a expresiei de fiabilitate folosind metoda relațiilor logice de semnal a fost completat de mine cu încă un pas (al 2-lea), menit să ducă la o expresie mai simplă a funcției de fiabilitate. Ca urmare, algoritmul are în principiu următorii pași:

1. Pornind de la nodul destinație ( $v_i$ ) se scriu succesiv relațiile logice de semnal conform tabelului 3.11.1. până când se ajunge la nodul sursă. Se face apoi substituția:

$$S(v_1) = S(\overline{v_1 \dots}) = 1 \quad (3.11.6)$$

deoarece semnalul este prezent în nodul sursă ( $v_1$ ) și în fuziunile de noduri care îl conțin pe  $v_1$ .

2. Se efectuează reducerea numărului de termeni ai expresiei sub forma disjunctivă pe baza relației booleene  $x_i Y \cup \bar{x}_i Y = Y$ . (Este de remarcat faptul că prin această operație nu dispăre caracterul de exclusivitate reciprocă a termenilor funcției rezultante, deoarece dacă  $x_i Y$  și  $\bar{x}_i Y$  au fost disjuncti cu ceilalți termeni ai funcției inițiale, atunci evident și reuniunea lor  $Y$  va păstra această proprietate. În schimb atrag atenția că utilizarea unei alte reguli a algebrei booleene, și anume  $x_i \cup \bar{x}_i Y = x_i \cup Y$  este interzisă, deoarece termenul  $Y$  obținut nu mai este disjunct cu  $x_i$ ).

3. În expresia obținută pentru  $S(v_i)$  se înlocuiesc variabilele logice cu probabilitățile corespunzătoare ( $x_i \rightarrow p_i$ ,  $\bar{x}_i \rightarrow q_i$ ), iar suma logică se transformă în sumă algebrică și se obține expresia fiabilității ( $s, t$ ). (Acest lucru este posibil deoarece termenii lui  $S(v_i)$  sunt reciproc disjuncti, și atunci probabilitatea reuniunii se poate transforma în sumă de probabilități).

**Exemplul 3.11.1.** Consider aici rețeaua de referință din [AGGA78], pentru a arăta că prin algoritmul îmbunătățit se obține o expresie de fiabilitate mai simplă decât cea dată în această sursă bibliografică. După efectuarea asignării sensurilor și a numerotării conform convenției TNC descrisă mai înainte, graful rețelei arată ca în figura 3.11.3.

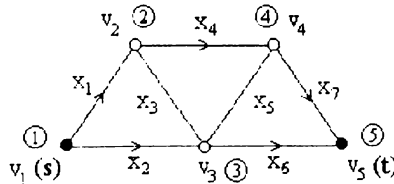
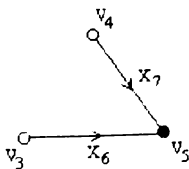


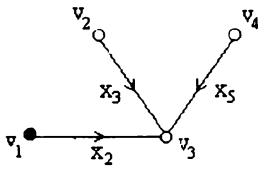
Figura 3.11.3.

Considerând toate nodurile perfecte, pe baza formulei nr.4 din tabelul 3.11.1 se poate scrie pentru nodul destinație:

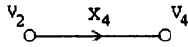


$$S(v_5) = x_6 \bar{x}_7 S(v_3) \cup \bar{x}_6 x_7 S(v_4) \cup x_6 x_7 S(\overline{v_3 v_4}) \quad (3.11.7)$$

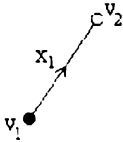
În continuare, aplicând relațiile nr.5 și nr.1 din tabelul 3.11.1, dezvolt  $S(v_3)$  până se ajunge la nodul sursă:



$$S(v_3) = x_2 \bar{x}_3 S(v_1) \cup \bar{x}_2 x_3 S(v_2) \cup x_2 x_3 S(\overline{v_1 v_2}) \cup \bar{x}_2 \bar{x}_3 x_5 S(v_4) \quad (3.11.8)$$

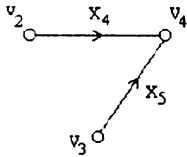


$$S(v_4) = x_4 S(v_2) \quad (3.11.9)$$

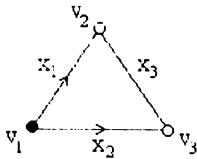


$$S(v_2) = x_1 S(v_1) \quad (3.11.10)$$

Analogue procedez cu \$S(v\_4)\$, pe baza relațiilor nr.4 și nr.6 din tabel:



$$S(v_4) = x_4 \bar{x}_5 S(v_2) \cup \bar{x}_4 x_5 S(v_3) \cup x_4 x_5 S(\overline{v_2 v_3}) \quad (3.11.11)$$

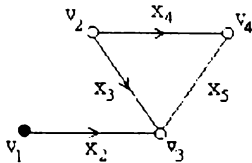


$$S(v_2) = x_1 S(v_1) \cup \bar{x}_1 x_2 x_3 S(v_1) \quad (3.11.12)$$

$$S(v_3) = x_2 S(v_1) \cup x_1 \bar{x}_2 x_3 S(v_1) \quad (3.11.13)$$

$$S(\overline{v_2 v_3}) = x_1 S(v_1) \cup \bar{x}_1 x_2 S(v_1) \quad (3.11.14)$$

Pentru \$S(\overline{v\_3 v\_4})\$ folosesc relațiile nr.7 și nr.1 din tabel:



$$S(\overline{v_3 v_4}) = (\bar{x}_2 \bar{x}_3 x_4 \cup \bar{x}_2 x_3) S(v_2) \cup x_2 \bar{x}_3 \bar{x}_4 S(v_1) \cup (x_2 \bar{x}_3 x_4 \cup x_2 x_3) S(\overline{v_1 v_2}) \quad (3.11.15)$$



$$S(v_2) = x_1 S(v_1) \quad (3.11.16)$$

Înlocuind acum expresiile (3.11.8) - (3.11.16) în expresia de bază (3.11.7) și ținând cont de (3.11.6), în urma simplificărilor permise de punctul 2 al algoritmului, se obține:

$$\begin{aligned}
 S(v_5) = & x_6 \bar{x}_7 (x_2 \cup x_1 \bar{x}_2 x_3 \cup x_1 \bar{x}_2 \bar{x}_3 x_4 x_5) \cup \bar{x}_6 x_7 (x_1 x_4 \cup \bar{x}_1 x_2 x_3 x_4 \bar{x}_5 \cup \\
 & \cup x_2 \bar{x}_4 x_5 \cup x_1 \bar{x}_2 x_3 \bar{x}_4 x_5 \cup \bar{x}_1 x_2 x_4 x_5) \cup x_6 x_7 (x_1 \bar{x}_2 \bar{x}_3 x_4 \cup \\
 & \cup x_1 \bar{x}_2 x_3 \cup x_2) = x_2 x_6 \bar{x}_7 \cup x_1 \bar{x}_2 x_3 x_4 \bar{x}_7 \cup x_1 \bar{x}_2 \bar{x}_3 x_4 x_5 x_6 \bar{x}_7 \cup \\
 & \cup x_1 x_4 \bar{x}_6 x_7 \cup \bar{x}_1 x_2 x_3 x_4 \bar{x}_5 \bar{x}_6 x_7 \cup x_2 \bar{x}_4 x_5 \bar{x}_6 x_7 \cup x_1 \bar{x}_2 x_3 \bar{x}_4 x_5 \bar{x}_3 x_7 \cup \\
 & \cup \bar{x}_1 x_2 x_4 x_5 \bar{x}_6 x_7 \cup x_1 \bar{x}_2 \bar{x}_3 x_4 x_6 x_7 \cup x_2 x_6 x_7
 \end{aligned} \quad (3.11.17)$$

Relația (3.11.17) fiind cu termeni disjuncti, aplicând pasul 3 al algoritmului rezultă direct expresia simbolică a fiabilității comunicației ( $s, t$ ):

$$\begin{aligned}
 R_{\tau} = & p_1 p_6 q_7 + p_1 q_2 p_3 p_6 q_7 + p_1 q_2 q_3 p_4 p_5 p_6 q_7 + p_1 p_4 q_6 p_7 + \\
 & + q_1 p_2 p_3 p_4 q_5 q_6 p_7 + p_2 q_4 p_5 q_6 p_7 + p_1 q_2 p_3 q_4 p_5 q_6 p_7 + \\
 & + q_1 p_2 p_4 p_5 q_6 p_7 + p_1 q_2 q_3 p_6 p_7 + p_1 q_2 p_3 p_6 p_7 + p_2 p_6 p_7
 \end{aligned} \quad (3.11.18)$$

Expresia (3.11.18) a fiabilității pentru exemplul considerat conține 11 termeni, față de 14 termeni rezultați ca urmare a metodei din [AGGA78].

□

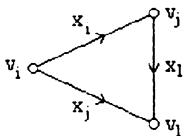
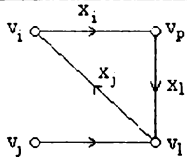
Pe baza expresiei (3.11.17) se pot obține căile minimale dintre nodurile  $v_1$  și  $v_5$  dacă se iau în considerare numai variabilele directe ale termenilor și se fac reducerile de termeni cu teorema absorbției ( $x_i Y \cup Y = Y$ ).

Dacă rețeaua conține noduri imperfecte, metoda relațiilor logice de semnal rămâne valabilă pentru calculul fiabilității. Este necesară însă modificarea formulelor din tabelul 3.11.1 prin înlocuirea lui  $x_i$  cu  $x_i V_i$  și a lui  $\bar{x}_i$  cu  $\bar{x}_i \cup x_i \bar{V}_i$ , dacă muchia  $x_i$  pornește din nodul  $v_i$  ( $V_i$  fiind variabilă logică asignată nodului  $v_i$ ,  $V_i = 1$  în caz de nod operațional).

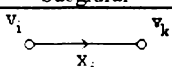
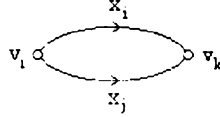
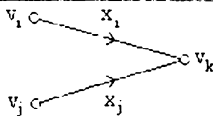
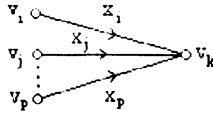
În analiza utilizării relațiilor logice de semnal am pornit de la o rețea modelată printr-un graf neorientat, în care unele muchii au fost direcționate știind că semnalul se transmite de la sursă spre destinație. Se poate însă pune întrebarea: cum se utilizează aceste relații pentru calculul fiabilității între perechea de noduri ( $s, t$ ) în cazul unei rețele presupusă a fi de la început orientată? Voi da în continuare răspuns la această întrebare. Cum am arătat, principiul relațiilor logice de semnal constă în descrierea cu termeni disjuncti a conectivității unui nod cu cele precedente vecine (conform ordinii de numerotare) în sensul ca de la acestea să ajungă cel puțin pe o cale semnalul la nodul în cauză. Dacă muchiile sunt orientate, atunci în expresie vor intra doar variabilele aferente muchiilor incidente către acest nod, funcționalitatea sau nefuncționalitatea celor incidente spre exterior neinfluențând ajungerea semnalului la nodul considerat. Ca urmare, dacă se dorește determinarea fiabilității de tip ( $s, t$ ) pentru un graf orientat, etapele expuse la algoritmul dinaintea exemplului 3.11.1 rămân valabile, iar formulele din tabelul 3.11.1 se utilizează cu următoarele observații:

- formulele nr. 1 - nr.5 sunt valabile, dar se aplică numai pentru muchiile incidente către nodul pentru care se exprimă relația de semnal
- formulele de la cazul nr.6 se modifică în funcție de sensul muchiei dintre  $v_i$  și  $v_j$
- formula nr.7 este valabilă, dar dacă sensul muchiei dintre  $v_i$  și  $v_j$  este invers, atunci se modifică corespunzător lui  $x_j = 0$

Rezultatele obținute și verificate pe diagrame Karnaugh pentru cazul rețelor orientate sunt trecute în tabelul 3.11.4.

Nr.	Subgraful	Relația logică de semnal
1-5, 7.	Ca la tabelul 3.11.1.	Ca la tabelul 3.11.1.
6.		$S(v_j) = x_i S(v_i)$ $S(v_i) = x_j S(v_i) \cup x_i \bar{x}_j x_i S(v_i)$ $S(\overline{v_i v_j}) = x_i S(v_i) \cup \bar{x}_i x_j S(v_i)$
7.		$S(\overline{v_p v_i}) = x_i \bar{x}_p S(v_i) \cup \bar{x}_i x_p S(v_j) \cup x_i x_p S(\overline{v_i v_j})$

Tabelul 3.11.4.

Nr.	Subgraful	Relația logică de semnal
1.		$S(v_k) = x_i S(v_i)$
2.		$S(v_k) = (x_i \cup \bar{x}_i x_j) S(v_i)$
3.		$S(v_k) = (x_i \cup \bar{x}_i x_j) S(v_i) S(v_j)$
4.		$S(v_k) = (x_i \cup \bar{x}_i x_j \cup \dots \cup \bar{x}_i \bar{x}_j \dots x_p) S(v_i) S(v_j) \dots$ $\dots S(v_j)$

Tabelul 3.11.5.

În cele ce urmează voi extinde metoda relațiilor logice de semnal pentru determinarea fiabilității globale într-o rețea orientată lipsită de cicluri. Dacă graful rețelei ar avea cicluri sau ar fi neorientat, atunci pentru problema fiabilității globale când nodul destinație nu mai este unic, nu se mai pot ordona nodurile conform convenției TNC și nu se mai poate urmări originea semnalului ajuns într-o buclă. Pentru graful neorientat, fiabilitatea globală (de tip SAT, source-to-all-terminal) înseamnă probabilitatea ca un semnal emis în nodul sursă să ajungă la toate celelalte noduri. De aceea, la scrierea relațiilor logice de semnal pentru nodurile grafului, luate în ordinea descrescătoare a

numerelor de ordine asociate, trebuie avut în vedere faptul că în fiecare nod obligatoriu trebuie să existe semnal. Rezultă că relația de semnal pentru un nod  $v_k$  va avea forma:

$$S(v_k) = (\text{condiția de conectivitate}) S(v_1) \dots S(v_j) \tag{3.11.19}$$

unde  $S(v_1) \dots S(v_j)$  exprimă condiția ca în toate nodurile vecine precedente să existe semnal, iar factorul din paranteză condiția ca acesta să ajungă pe o cale funcțională la  $v_k$ . Dacă expresia din paranteză se exprimă sub formă de sumă de termeni disjuncți, atunci relația se poate transpune direct într-o expresie de probabilitate. Pe această bază am întocmit formulele de calcul pentru cazurile uzuale de subrețele din tabelul 3.11.5, verificate pe diagrame Karnaugh, cu mențiunea că în relații nu trebuie să intre în calcul decât muchiile incidente către nodul considerat. Voi ilustra metoda propusă pe un exemplu.

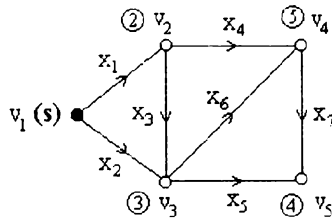
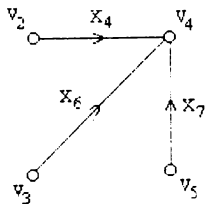
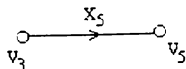


Figura 3.11.6.

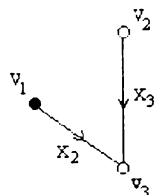
**Exemplul 3.11.2.** Pentru rețeaua orientată din figura 3.11.6 nodurile au fost ordonate conform procedurii descrise anterior, nodul  $v_1$  fiind sursa. Determinarea fiabilității SAT presupune aplicarea relațiilor nr.1, nr.3 și nr.4 din tabelul 3.11.5, pornind de la nodul  $v_4$ :



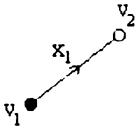
$$S(v_4) = \{x_4 \cup \bar{x}_4 x_6 \cup \bar{x}_4 \bar{x}_6 x_7\} S(v_2) S(v_3) S(v_5) \tag{3.11.20}$$



$$S(v_5) = x_5 S(v_3) \tag{3.11.21}$$



$$S(v_3) = \{x_2 \cup \bar{x}_2 x_3\} S(v_1) S(v_2) \tag{3.11.22}$$



$$S(v_2) = x_1 S(v_1) \quad (3.11.23)$$

Efectuând înlocuirile în (3.11.20) și ținând cont că  $S(v_1) = 1$ , se obține:

$$S(v_4) = x_1 x_5 (x_2 \cup \bar{x}_2 x_3) (x_4 \cup \bar{x}_4 x_6 \cup \bar{x}_4 \bar{x}_6 x_7) \quad (3.11.24)$$

Factorii acestei relații corespund la evenimente independente, iar în cadrul fiecărui factor termenii corespund la evenimente reciproc exclusive, deci se poate scrie direct expresia de fiabilitate:

$$R_{SAT} = p_1 p_5 (p_2 + q_2 p_3) (p_4 + q_4 p_6 + q_4 q_6 p_7) \quad (3.11.25)$$

□

Utilizarea relațiilor logice de semnal face posibilă determinarea expresiei de fiabilitate prin parcurgerea din aproape în aproape a unei rețele generale, fără a dispune de informații globale de genul căilor sau tăieturilor. Concepută inițial doar pentru fiabilitatea de tip terminal, metoda a fost extinsă și la cazul fiabilității globale.

### 3.12. Importanța legăturilor în contextul fiabilității rețelei

Fiabilitatea de tip terminal sau multiterminal a unei rețele modelate printr-un graf probabilistic  $G$ , în care nodurile sunt perfect fiabile, se exprimă ca o funcție de fiabilitățile  $p_i$  ale legăturilor:

$$R(G) = f(p_1, p_2, \dots, p_n) \quad (3.12.1)$$

Valoarea acestei funcții, pentru un graf cu  $n$  muchii, depinde de valoarea fiabilității fiecărei muchii, dar și de modul lor de dispunere, adică de topologia rețelei. Importanța fiabilității unei legături este o măsură cantitativă a contribuției legăturii respective la fiabilitatea rețelei. Ea se definește ca și rata cu care crește fiabilitatea rețelei la creșterea fiabilității muchiei. La limită se obține *importanța marginală a fiabilității* (MRI, marginal reliability importance) pentru muchia  $e_i$ , dată de:

$$I_{\frac{1}{2}}(i) = \frac{\partial R(G)}{\partial p_i} \quad (3.12.2)$$

În mod similar se poate defini *importanța marginală a defectării* (MFI, marginal failure importance) cu referire la funcția de nefiabilitate  $F(G) = 1 - R(G)$ , ca fiind

$$I'_{\frac{1}{2}}(i) = \frac{\partial F(G)}{\partial q_i} \quad (3.12.3)$$

unde  $q_i = 1 - p_i$ .

Valoarea lui  $I_{\frac{1}{2}}(i)$  este situată în intervalul  $[0, 1]$ , iar a lui  $I'_{\frac{1}{2}}(i)$  în  $[ -1, 0]$ . Este evident că între cele două mărimi există egalitatea:



$$I_G(i) = -I'_G(i) \quad (3.12.4)$$

Mărimile MRI și MFI pot fi exprimate și fără derivare pe baza funcțiilor de fiabilitate (nefiabilitate) ale grafurilor  $G^*e_i$  și  $G-e_i$ , obținute din  $G$  cu muchia  $e_i$  contractată (perfectă), respectiv ștearsă (defectă). Relațiile de calcul sunt date de următoarea leamnă:

**Lema 3.12.1.** Într-un graf  $G(V,E)$  neorientat, pentru MRI, respectiv MFI a unei muchii  $e_i$ , sunt adevărate relațiile:

$$I_G(i) = R(G^*e_i) - R(G-e_i) \quad (3.12.5)$$

$$I'_G(i) = F(G^*e_i) - F(G-e_i) \quad (3.12.6)$$

*Demonstrație.* Dezvoltând  $R(G)$  după muchia  $e_i$ , se obține:

$$\begin{aligned} R(G) &= p_i R(G|x_i = 1) + (1-p_i) R(G|x_i = 0) = \\ &= p_i R(G^*e_i) + (1-p_i) R(G-e_i) \end{aligned} \quad (3.12.7)$$

Efectuând derivarea conform (3.12.2) rezultă relația căutată. Demonstrația este similară și pentru (3.12.6). □

Hong și Lie au extins mărimea de importanță a fiabilității unei legături la importanța a două muchii într-o rețea neorientată [HONG93]. *Importanța reunită a fiabilității* (JRI, joint reliability importance) pentru două muchii  $e_i$  și  $e_j$  este dată de

$$I_G(i, j) = \frac{\partial^2 R(G)}{\partial p_i \partial p_j} \quad (3.12.8)$$

iar *importanța reunită a defectării* (JFI, joint failure importance) de:

$$I'_G(i, j) = \frac{\partial^2 F(G)}{\partial q_i \partial q_j} \quad (3.12.9)$$

Se arată ușor, în urma unei dezvoltări similare cu cea de la lema 3.12.1, dar pentru ambele muchii, că este adevărată relația:

$$I_G(i, j) = I_{G^*e_i}(i) - I_{G-e_i}(i) \quad (3.12.10)$$

Este clar că  $-1 \leq I_G(i, j) \leq 1$ . Egalitatea (3.12.10) dă posibilitatea efectuării următoarelor interpretări:

- a) Dacă  $I_G(i, j) \leq 0$ , atunci  $e_i$  este mai importantă când  $e_j$  este defect, decât dacă  $e_j$  este funcțional. muchiile având un caracter de legare în paralel.
- b) Dacă  $I_G(i, j) \geq 0$ , atunci  $e_i$  este mai importantă când  $e_j$  este funcțional, decât dacă  $e_j$  este defect, muchiile având un caracter de legare în serie.

În funcție de semnul lui  $I_G(i, j)$ , perechile de muchii ale unui graf pot fi clasificate în trei grupe:

- perechi care nu sunt conținute în nici o cale (pentru care  $I_G < 0$ )
- perechi care nu sunt conținute în nici o tăietură (pentru care  $I_G \geq 0$ )
- perechi conținute atât în căi, cât și în tăieturi (pentru care JRI poate fi atât pozitiv, cât și negativ, în funcție de valoarea fiabilităților individuale)

**Exemplul 3.12.2.** Pentru rețeaua punte din figura 3.7.1, cele trei clase cu referire la fiabilitatea comunicației  $(s, t)$  sunt:

$$C_1 = \{(e_1, e_2), (e_4, e_5)\}$$

$$C_2 = \{(e_1, e_4), (e_2, e_5)\}$$

$$C_3 = \{(e_1, e_3), (e_2, e_3), (e_3, e_4), (e_3, e_5), (e_1, e_5), (e_2, e_4)\}$$

□

Contribuția fiecărei legături la valoarea fiabilității rețelei este în general diferită, de aceea se pune problema de a găsi un criteriu de ordonare a legăturilor după importanța acestora. Un astfel de criteriu este cel care pornește de la expresia polinomială a fiabilității  $R(G)$ , considerând că toate muchiile au probabilități egale de defectare ( $p$ ). Pentru fiabilitatea cea mai generală, de tip  $K$ -terminal, expresia acestui polinom este:

$$R(G) = \sum_{i=1}^l a_i p^i (1-p)^{l-i} \quad (3.12.11)$$

unde  $a_i$  reprezintă numărul de subgrafuri distincte  $G_S$  ale lui  $G$  cu  $i$  muchii având proprietatea că fiecare pereche de noduri țintă este unită de o cale formată din muchii care aparțin lui  $G_S$ . Acest polinom este invariant pentru un anumit graf, ceea ce nu necesită cunoașterea prealabilă a fiabilității componentelor pentru a analiza importanța relativă a legăturilor. Comparând două muchii  $e_i$  și  $e_j$  din punct de vedere al importanței acestora în fiabilitatea rețelei, se consideră conform [PAGE94] că  $e_i$  este mai importantă decât  $e_j$  ( $e_i > e_j$ ) dacă pentru orice  $p \in (0, 1)$  sunt satisfăcute relațiile:

$$R(G-e_i) \leq R(G-e_j) \quad (3.12.12)$$

și

$$R(G^*e_i) \geq R(G^*e_j) \quad (3.12.13)$$

Dacă o muchie apare mai critică în relația (3.12.12), iar cealaltă în relația (3.12.13), atunci ele nu sunt comparabile. De asemenea, muchiile nu se pot ordona dacă funcțiile de fiabilitate corespunzătoare se intersectează în intervalul  $(0, 1)$ . Din cauza acestor posibilități, criteriile introduse duc numai la o ordonare parțială a legăturilor. Dacă în cele două relații se verifică egalitatea, atunci  $e_i \approx e_j$ .

În sinteza unei rețele, dacă se alocă probabilități egale tuturor legăturilor, atunci modificarea fiabilității unei legături mai importante va duce la o modificare mai accentuată a fiabilității rețelei decât schimbarea fiabilității unei legături mai puțin importante. Întrucât în timp fiabilitatea unei legături se poate modifica doar în sens descrescător, consider că este mai semnificativă luarea în considerare a relației (3.12.12) la studiul comparativ a două muchii. Faptul că muchia  $e_i$  să cadă (și deci graful  $G$  să se transforme în  $G-e_i$ ) este

o situație real posibilă, pe când contractarea ( $e_i$  să devină perfectă) este imposibilă. De aceea, am căutat răspunsul în [KAKAS95b] la întrebarea: ce modificare suferă fiabilitatea rețelei după căderea unei legături? Evident, ar fi de dorit ca  $R(G-e_i) = R(G)$ , ceea ce este posibil însă numai dacă muchia  $e_i$  este irelevantă ( $\frac{\partial R(G)}{\partial p_i} = 0$ ). Ceea ce poate reprezenta

însă un deziderat în sinteză este ca fiabilitatea rețelei să se reducă cu aceeași valoare la căderea oricărei legături. Această condiție este aplicabilă la toate muchiile numai pentru anumite topologii particulare, de aceea consider doar o submulțime a muchiilor:

$$\Delta R = p_i \frac{\partial R(G)}{\partial p_i} = p_j \frac{\partial R(G)}{\partial p_j} = p_k \frac{\partial R(G)}{\partial p_k} = \dots \quad (3.12.14)$$

Egalitățile date de (3.12.14) împreună cu o valoare impusă pentru  $R(G)$ , formează un sistem de ecuații în care necunoscute sunt fiabilitățile individuale  $p_1, p_2, p_3, \dots$  ale componentelor. Sistemul fiind neliniar, rezolvarea este de regulă dificilă și nu are totdeauna soluție (ca rădăcinile să fie cuprinse în intervalul (0,1)). Subliniez că acest criteriu de a aloca fiabilități nu se poate aplica la orice muchie. Astfel, dacă muchia este în serie cu tot restul grafului  $G$ , atunci defectarea ei duce la fiabilitate zero pentru toată rețeaua, ceea ce nu poate constitui un țel și pentru celelalte muchii.

**Exemplul 3.12.3.** Ca exemplu consider graful  $G$  din figura 3.12.1.a, în care studiez fiabilitatea comunicației între nodurile  $v_1$  și  $v_5$ , dată de funcția:

$$R(G) = p_2 p_5 + p_1 p_4 p_6 + p_1 p_3 p_5 + p_2 p_3 p_4 p_6 - p_1 p_2 p_3 p_5 - p_2 p_3 p_4 p_5 p_6 - p_1 p_3 p_4 p_5 p_6 - p_1 p_2 p_4 p_5 p_6 - p_1 p_2 p_3 p_4 p_6 + 2 p_1 p_2 p_3 p_4 p_5 p_6 \quad (3.12.15)$$

Pentru probabilități egale se obține polinomul:

$$R(G) = p^2(1-p)^4 + 6p^3(1-p)^3 + 12p^4(1-p)^2 + 6p^5(1-p) + p^6 = p^2 + 2p^3 - 4p^4 + 2p^6 \quad (3.12.16)$$

Pentru acest polinom se poate aplica criteriul de ordonare dat de (3.12.12) și (3.12.13) pentru fiecare pereche de muchii (Anexa 3). Rezultatul este reprezentat sub formă de diagramă Hasse în figura 3.12.1.b, pe care se constată că muchia  $e_5$  are contribuția cea mai mare la fiabilitatea rețelei, iar muchia  $e_3$  este cea mai puțin importantă. Muchiile  $e_4$  și  $e_5$  sunt echivalente din acest punct de vedere. Dacă se ia  $p = 0.8$  pentru toate muchiile, atunci fiabilitatea rețelei este  $R(G) = 0.88$ . Reducerea de fiabilitate cauzată de căderea muchiei  $e_3$  este nesemnificativă,  $R(G-e_3) = 0.82$ , însă dacă se elimină  $e_5$  atunci diminuarea este considerabilă  $R(G-e_5) = 0.59$ .

Mi-am propus ca, menținând  $R(G) = 0.88$ , să calculez probabilitățile care trebuie alocate muchiilor  $e_1$  și  $e_2$  pentru ca prin căderea oricăreia dintre ele,  $R(G)$  să se diminueze cu aceeași valoare (restul muchiilor păstrează  $p = 0.8$ ). Rezolvând sistemul de ecuații (3.12.14) și (3.12.15) cu necunoscutele  $p_1$  și  $p_2$ , am obținut  $p_1 = 0.83$  și  $p_2 = 0.78$ . Concluzia este interesantă: trebuie alocată o fiabilitate mai mare muchiei mai puțin importante ( $e_1$ ), deoarece fiabilitatea mai mică alocată lui  $e_2$  este compensată de poziția ei topologică în rețea.

□

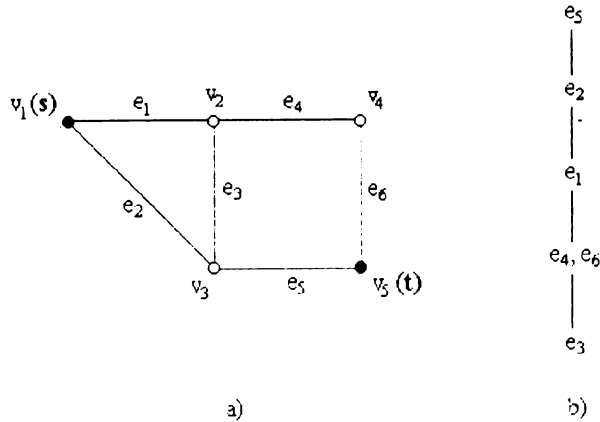


Figura 3.12.1.

Din cele prezentate s-a putut constata că fiabilitatea bazată pe existența unei conectivități între o submulțime de noduri dintr-o rețea depinde atât de topologia rețelei, cât și de fiabilitățile individuale ale legăturilor. Defectarea unei legături poate duce la reducerea substanțială a indicatorului global de fiabilitate. Inegalitățile de cauză topologică pot fi compensate în anumite situații prin alegerea convenabilă a fiabilităților asignate componentelor.

### 3.13. Sisteme cu defecțiuni dependente

În general metodele de evaluare a fiabilității presupun că defecțiunile componentelor sunt statistic independente (*s-independent*). De cele mai multe ori această presupunere nu este însă realistă, deoarece componentele alăturate sunt probabil afectate de aceleași condiții de mediu (de exemplu interferența, bruiatul sau perturbațiile atmosferice pot afecta simultan mai multe legături de comunicație), mai multe componente pot depinde de aceeași resursă fizică comună (sursă de tensiune, linie de transmisie distribuită) sau defectarea unei componente poate duce la defectarea unei alte componente. Spragins a arătat că analiza de fiabilitate bazată pe ipoteza defectelor independente poate duce la rezultate periculoase de optimiste [SPRA86].

Există mai multe încercări de studiu a fiabilității sistemelor distribuite cu componente interdependente. Un mod de abordare este acela de a specifica dependența statistică dintre componentele sistemului prin *probabilități condiționate de defectare*, astfel încât probabilitatea comună de defectare a două sau mai multe componente poate fi evaluată cu ajutorul relației de dezvoltare în lanț. Această abordare (modelul  $q$ - $\psi$ , modelul E) are însă dezavantajul că numărul de termeni ai expresiei de probabilitate condiționată crește exponențial cu numărul de componente și în plus termenii, cu număr mare de parametri, trebuie să satisfacă cerințele de consistență.

O abordare complet diferită stă la baza *modelului de fiabilitate bazat pe evenimente* (EBRM, Event-Based Reliability Model) dezvoltat de Lam și Li pentru rețele de comunicație în care defecțiunile liniilor sunt statistic dependente [LAM86]. Metoda încearcă identificarea și modelarea explicită a evenimentelor care cauzează căderea

legăturilor. Evenimentele considerate fiind independente, se elimină necesitatea utilizării probabilităților condiționate.

Modelul presupune că într-o rețea există evenimente care cauzează căderea simultană a mai multor linii de comunicație. Rețeaua se reprezintă ca de obicei printr-un graf, iar evenimentele care produc defecte se reprezintă prin "elemente-eveniment" ("event elements"), care se adaugă la muchiile afectate (figura 3.13.1). Un element-eveniment se spune că este în modul "jos" atunci când apare evenimentul cauză al defectului corăspunzător, respectiv în modul "sus" în rest. Dacă un element-eveniment este în modul "jos", atunci toate legăturile afectate vor cădea. Toate evenimentele se consideră a fi independente și apar cu probabilități cunoscute. Astfel se modelează defecțiuni dependente cu ajutorul unor elemente-eveniment independente, ceea ce duce la simplificarea calculului și permite utilizarea unor algoritmi cunoscuți de evaluare a fiabilității. De exemplu, dacă pentru figura 3.13.1 se cere probabilitatea de a exista o comunicație între  $s$  și  $t$ , alegând metoda incluziunii-excluziunii (sau un alt procedeu) se obține:

$$P_{st} = \Pr(e_1) + \Pr(e_2e_3) - \Pr(e_1e_2e_3) \quad (3.13.1)$$

unde termenii reprezintă probabilitățile ca legăturile din paranteze să fie operaționale. Probabilitatea ca un grup de legături să funcționeze este dată de probabilitatea ca toate elementele-eveniment implicate să fie în modul "sus". Având în vedere că evenimentele sunt independente, expresia de calcul devine:

$$\begin{aligned} P_{st} &= \Pr(1\ 2\ 3) + \Pr(1\ 3\ 4\ 5\ 6) - \Pr(1\ 2\ 3\ 4\ 5\ 6) = \\ &= p_1 p_2 p_3 + p_1 p_3 p_4 p_5 p_6 - p_1 p_2 p_3 p_4 p_5 p_6 \end{aligned} \quad (3.13.2)$$

unde  $p_i$  reprezintă probabilitatea ca elementul-eveniment  $i$  să fie în modul "sus". Se observă că nu se ridică probleme de consistență.

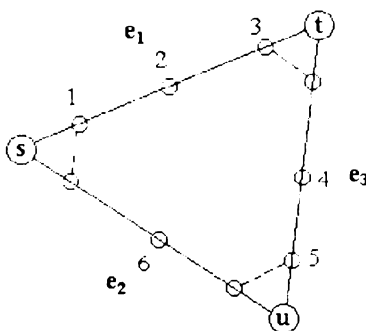


Figura 3.13.1.

Modelul de fiabilitate EBRM are și avantajul că permite o înțelegere mai bună a cauzelor și efectelor defecțiunilor din rețele, fiind flexibil și adaptabil la modificări în condițiile de lucru ale sistemului. Identificarea parametrilor elementelor-eveniment este o problemă practică care ține de cunoașterea echipamentelor utilizate. Veeraraghavan și Trivedi au extins modelul EBRM și la sistemele cu stări multiple [VEER93][VEER94].

Heffes și Kumar analizează problema supraviețuirii rețelelor cu noduri vulnerabile dependente prin prisma conectivității grafului care descrie topologia rețelei [HEFF86]. Prin analogie cu conectivitatea de nod (numărul minim de noduri a căror eliminare duce la deconectarea grafului) se introduce noțiunea de conectivitatea generalizată (generalized connectivity) ca fiind numărul minim de evenimente care produc defecte, necesare pentru a deconecta graful. Se presupune că un eveniment poate duce la distrugerea mai multor noduri.

Pentru a modela posibilitățile prin care aceste evenimente afectează nodurile se folosește un *graf de dependență* (dependence graph), care se construiește în urma identificării evenimentelor astfel încât nodurile neadiacente ale acestuia nu pot fi distruse de același eveniment. În figura 3.13.2 se prezintă un exemplu de graf de dependență  $D$  pentru graful topologic  $G$  al rețelei. Conectivitatea logică (în sensul obișnuit) a lui  $G$  este 4, dar în prezența interdependenței între noduri, aceasta nu este o măsură corectă a conectivității fizice (în sensul generalizat). Din interpretarea grafului de dependență rezultă că nodurile distruse de același eveniment trebuie să fie complet conectate în acest graf, ele formând un *grup admisibil* (admissible clique) de noduri. Nu orice grup complet conectat este însă și admisibil, numai dacă există un eveniment capabil să distrugă tocmai toate nodurile din acest grup. Un set de grupuri admisibile  $\mathcal{S}$  care realizează și o partiționare a lui  $D$ , constituie o partiție de grupuri admisibile.

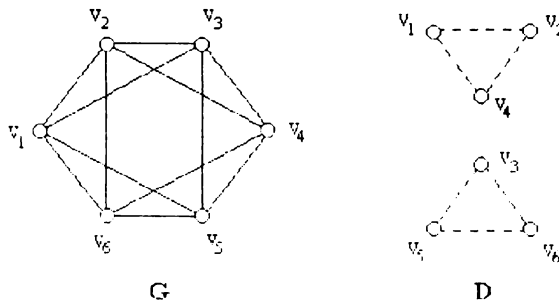


Figura 3.13.2.

Modelul de dependență prezentat permite studiul a două subclase, și anume dependența puternică și dependența slabă. La modelul de dependență puternică (strong dependence) se presupune că două noduri adiacente în graf trebuie distruse împreună, pe când la dependența slabă (weak dependence) ele pot fi distruse împreună, dar și separat. Evident că tipul de dependență considerat are implicații asupra gradului de conectivitate generalizat. Deoarece orice partiție de grupuri admisibile a lui  $D$  corespunde unei posibilități de a distruge toate nodurile lui  $G$ , rezultă că conectivitatea generalizată a lui  $G$  în raport cu  $D$  nu este mai mare decât cardinalitatea partiției minime de grupuri admisibile a lui  $D$ :

$$\omega(G, D) \triangleq \min \{ \text{card } \mathcal{S} \mid \mathcal{S} \text{ este partiție a lui } D \} \quad (3.13.3)$$

Pentru calculul lui  $\omega(G, D)$  în cazul unei dependențe puternice se prezintă în [HEFF86] o modalitate bazată pe acoperirea minimă în graful  $G$ . În cazul unei dependențe slabe acest calcul este o problemă foarte dificilă, de aceea se determină numai niște limite extreme.

Determinarea unei limite inferioare de fiabilitate în cazul unor defecțiuni dependente constituie un obiectiv și pentru un model probabilistic [CARR86]. Se caută o margine inferioară pentru fiabilitatea globală de tip multiterminal a rețelei, atunci când despre dependența statistică a defectelor de legătură este disponibilă numai o informație restrânsă.

În cazul în care nu se cunoaște nimic despre dependența dintre defecte, se admite ipoteza cea mai pesimistă care duce la o limită (inferioară) absolută a fiabilității, numită limită de ordinul întâi (first-order bound). Determinarea se face selectând cel mai fiabil arbore de acoperire ( $T$ ) al grafului rețelei, deoarece fiabilitatea rețelei este cel puțin egală cu fiabilitatea acestui arbore. Defectarea acestui arbore se produce la defectarea ( $\bar{e}$ ) a cel puțin uneia din muchiile componente  $e \in T$ , deci pe baza formulei de calcul a probabilității de defectare rezultă o limită superioară pentru aceasta:

$$\Pr\left(\bigcup_{e \in T} \bar{e}\right) \leq \sum_{e \in T} \Pr(\bar{e}) = \sum_{e \in T} q_e \quad (3.13.4)$$

unde  $q_e$  reprezintă probabilitatea de defectare a muchiei  $e$ . Dacă se consideră că pentru fiecare muchie există o valoare minimă de fiabilitate  $a_e \leq p_e$ , unde  $p_e = 1 - q_e$ , deci  $q_e \leq \bar{a}_e$ , atunci se obține o limită inferioară de fiabilitate a rețelei, dată de relația:

$$L = 1 - \sum_{e \in T} \bar{a}_e = 1 - \sum_{e \in T} (1 - a_e) \quad (3.13.5)$$

S-a constatat însă că acest calcul dă o valoare foarte mică a limitei, practic inutilizabilă.

O îmbunătățire a limitei inferioare de fiabilitate se obține dacă se cunosc informații despre probabilitățile de defectare ale perechilor de muchii. Astfel se obține o limită de ordinul al doilea (second-order bound), care se poate determina pe baza unui raționament similar cu cel de mai înainte. În urma testelor efectuate pe diferite rețele a rezultat clar că această limită este în mod semnificativ mai mică decât o limită care presupune independența statistică între defecte.

### 3.14. Fiabilitatea sistemelor cu stări multiple

Majoritatea metodelor tradiționale de evaluare a fiabilității sistemelor cu componente multiple lucrează cu componente presupuse a avea două stări: funcțional sau defect (nefuncțional). Această abordare simplifică o mulțime de situații reale, în care sistemul și componentele sale pot opera într-o întreagă gamă de performanțe, variind între funcționarea perfectă și căderea completă. De exemplu, o linie de transmisie digitală poate fi expusă la erori datorită perturbațiilor electromagnetice. Pachetele de date trebuie retransmise, deci scade capacitatea de comunicație a liniei, dar ea nu se desființează. Ca urmare este oportună luarea în considerare a mai multor stări posibile ale sistemului.

**Definiția 3.14.1.** Un sistem ale cărui componente se consideră că pot avea mai mult de două stări se numește *sistem cu stări multiple* (multistate system).

Găsirea de algoritmi eficienți pentru calculul fiabilității sistemelor cu stări multiple este foarte importantă, deoarece aceștia pot fi mari consumatori de timp. Modelul sistemelor cu stări multiple se poate construi prin analogia cu sistemele binare (cu două stări) [WOOD85]. Astfel componentele și sistemele cu mai multe stări vor putea fi analizate cu ajutorul algoritmilor binari existenți. Pentru a realiza această corespondență și

a putea descrie starea unei componente, respectiv a sistemului, se introduc variabilele binare  $x_{ij}$  și  $\Phi^k(X)$ . Fie un sistem cu  $n$  componente, starea unei componente fiind indicată de valoarea lui  $x_i$ , deci  $x_i \in \{0, 1, \dots, N_i\}$ , unde  $N_i$  reprezintă cea mai bună stare a componentei  $i$ . Variabila binară  $x_{ij}$  semnifică starea componentei  $i$  raportată la o stare  $j$ , adică:

$$\begin{aligned} x_{ij} &= 0 && \text{dacă } x_i < j \\ x_{ij} &= 1 && \text{dacă } x_i \geq j \end{aligned}$$

Starea componentei  $i$  la un moment dat va putea fi calculată în consecință cu relația:

$$x_i = \sum_{j=1}^{N_i} x_{ij} \quad (3.14.1)$$

Vectorul  $X = (x_1, \dots, x_n)$  reprezintă vectorul de stare al componentelor din sistem. Starea sistemului la un moment dat se exprimă prin funcția de structură a sistemului  $\Phi(X)$ , care poate lua valori în intervalul  $\Phi(X) \in \{0, 1, \dots, M\}$ , unde  $M$  reprezintă cea mai bună stare a sistemului.

**Definiția 3.14.2.** Funcția de structură a unui sistem cu  $n$  componente cu stări multiple este funcția  $\Phi: \{0, 1\}^n \rightarrow \{0, 1, \dots, M\}$  definită prin expresia

$$\Phi(X) = \sum_{k=1}^M \Phi^k(X) \quad (3.14.2)$$

în care variabila binară  $\Phi^k(X)$  indică starea sistemului în raport cu o stare  $k$ , adică:

$$\begin{aligned} \Phi^k(X) &= 0 && \text{dacă } \Phi(X) < k \\ \Phi^k(X) &= 1 && \text{dacă } \Phi(X) \geq k \end{aligned}$$

Prin acest model, funcția de structură cu stări multiple  $\Phi(X)$  devine o funcție monotonă, ceea ce înseamnă că deteriorarea unei componente nu va duce niciodată la îmbunătățirea stării sistemului. Dezavantajul inerent este însă faptul că se pierde  $s$ -independența dintre componente. Chiar dacă componentele originale cu stări multiple sunt  $s$ -independente, componentele binare  $x_{ij}$  care le înlocuiesc în model sunt în general corelate, deoarece  $x_{ij} = 1$  implică  $x_{ik} = 1$  pentru toate  $k \leq j$ . Funcțiile de structură binare sunt de asemenea corelate, deoarece  $\Phi^k(X) = 1$  implică  $\Phi^l(X) = 1$  pentru toate  $k \leq l$ .

Fiabilitatea sistemului se determină pentru fiecare nivel  $k$ , ca fiind probabilitatea ca sistemul să se afle cel puțin în starea  $k$ :

$$R(k) = \Pr\{\Phi(X) \geq k\} = \Pr\{\Phi^k(X) = 1\} = E[\Phi^k(X)] \quad (3.14.3)$$

Această mărime se numește *fiabilitatea la nivelul  $k$*  (reliability at level  $k$ ). Se poate calcula ușor și o altă mărime, numită *eficiența sistemului* (system effectiveness), dată de relația:

$$E[\Phi(X)] = \sum_{k=1}^M E[\Phi^k(X)] \quad (3.14.4)$$



Tehnica de modelare descrisă permite folosirea metodelor uzuale de genul schemelor bloc sau arborilor de defectare pentru calculul fiabilității sistemelor cu stări multiple.

**Exemplul 3.14.3.** Pentru ilustrare se consideră un sistem cu două calculatoare interconectate, fiecare cu câte o memorie proprie. Sistemul este funcțional dacă un procesor dispune de întreaga sa memorie și celălalt cel puțin de jumătate. Sistemul este operațional, dar cu performanțe reduse (fail-soft), dacă numai un procesor lucrează, cu întreaga sa memorie, sau ambele pot funcționa cu câte o jumătate din capacitatea de memorie. Sistemul este nefuncțional, dacă un procesor are numai jumătate din memoria sa disponibilă, iar celălalt deloc. Mulțimea stărilor celor două componente va fi

$$x_1, x_2 \in \{0 - \text{fără memorie}, 1 - \text{cu jumătate de memorie}, 2 - \text{cu întreaga memorie}\}$$

iar mulțimea stărilor sistemului:

$$\Phi(X) \in \{0 - \text{nefuncțional}, 1 - \text{performanțe reduse}, 2 - \text{funcțional}\}$$

Stările sistemului pentru toate combinațiile posibile ale stărilor componentelor sunt următoarele:

$$\Phi(2,2) = \Phi(2,1) = \Phi(1,2) = 2 \quad (3.14.5.a)$$

$$\Phi(0,2) = \Phi(2,0) = \Phi(1,1) = 1 \quad (3.14.5.b)$$

$$\Phi(0,1) = \Phi(1,0) = \Phi(0,0) = 0 \quad (3.14.5.c)$$

Funcția de structură care exprimă situațiile în care sistemul se află cel puțin în starea 1 rezultă ca fiind:

$$\Phi^1(X) = \max(x_{12}, x_{22}, x_{11}, x_{21}) \quad (3.14.6)$$

adică afirmația este adevărată dacă  $x_1$  este 2, sau  $x_2$  este 2, sau  $x_1 \geq 1$  și  $x_2 \geq 1$ . În mod similar:

$$\Phi^2(X) = \max(x_{11}, x_{22}, x_{12}, x_{21}) \quad (3.14.7)$$

Pentru determinarea fiabilităților corespunzătoare se pot construi schemele bloc (block diagrams) din figura 3.14.1, compuse din structuri de tip serie și paralel. În fiecare bloc se află înscrise numărul componente și starea minimă necesară acelei componente. Fiabilitatea la nivelul  $k$  rezultă din calculul lui  $\Pr\{\Phi^k(X) = 1\}$  pe baza schemei, utilizând regulile cunoscute de componere serială și paralelă. Dacă componentele sunt  $s$ -independente și pentru orice componentă  $i$  apare în schemă cel mult o dată o variabilă binară  $x_i$ , atunci fiabilitatea se calculează simplu înlocuind pe  $x_i$  cu probabilitatea corespunzătoare. În caz contrar, cum este și exemplul considerat, este necesară dezvoltarea de probabilitate condiționată:

$$R(1) = E[\Phi^1(X)] = p_{12} E[\Phi^1(X) | x_{12} = 1] + (1-p_{12}) E[\Phi^1(X) | x_{12} = 0] \quad (3.14.8)$$

În această expresie  $p_{12}$  reprezintă probabilitatea ca componenta 1 să se afle cel puțin în starea 2, în general:

$$p_{ij} = \Pr\{x_{ij} = 1\}$$

□

Similar cu extinderea schemelor bloc se poate face și extinderea analizei fiabilității sistemelor cu stări multiple prin arbori de defectare. Arborele de defectare (fault tree) și arborele de eveniment (event tree) sunt scheme logice cu porți ȘI și SAU care constau dintr-un eveniment de vârf și o structură de indicare a căilor care fac posibilă producerea evenimentului. Termenul de arbore de defectare se folosește atunci când evenimentul din vârf este căderea sistemului, iar arborele de eveniment în cazul succesului. La sistemele cu stări multiple câte un astfel de arbore trebuie construit pentru fiecare stare a sistemului. În figura 3.14.2 se dau arborii de reușită pentru exemplul anterior. Și aici, dacă o componentă apare de mai multe ori într-un arbore, pentru calculul fiabilității trebuie utilizate probabilități condiționate.

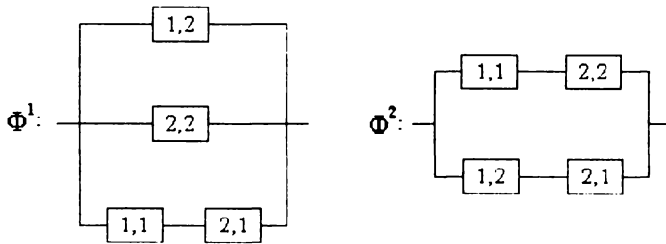


Figura 3.14.1.

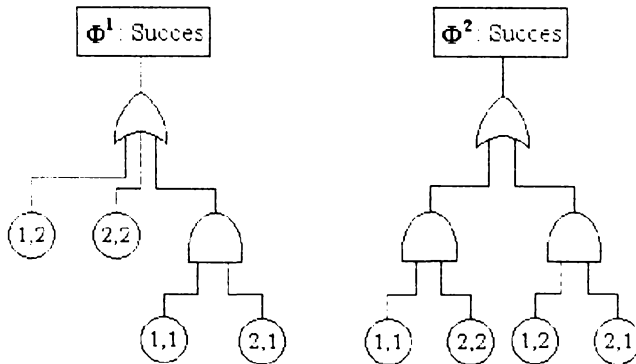


Figura 3.14.2.

În [AVEN85] se prezintă alți doi algoritmi eficienți pentru calculul fiabilității, respectiv non-fiabilității la nivelul  $k$  a sistemelor monotone cu stări multiple cu componente  $s$ -independente. Algoritmii se bazează pe metoda de decompoziție a lui Doulliez și Jamoulle.

Primul algoritm, care calculează fiabilitatea  $R(k)$  la nivelul  $k$ , pornește de la necesitatea cunoașterii vectorilor de cale minimală (minimal path vectors). Un vector de stare  $Y$  al componentelor reprezintă o cale minimală dacă  $\Phi(Y) \geq k$ , dar pentru orice  $X < Y$

se obține  $\Phi(X) < k$ . Prin comparare cu acești vectori minimali, spațiul stărilor  $\mathcal{S} = \{X\}$  se descompune în următoarele seturi de stări sigure:

- setul stărilor acceptabile, pentru  $\Phi(X) \geq k$
- setul stărilor neacceptabile, pentru  $\Phi(X) < k$
- setul stărilor nedefinite

Fiecare set de stări nedefinite se descompune din nou, până când nu va mai rămâne nici un astfel de set. Fiecare set de stări nedefinite  $S$  are asociați doi vectori  $b^0$  și  $b$ , care reprezintă stările limită, astfel încât:

$$S = \{X \in \mathcal{S}; b \leq X \leq b^0\} \quad (3.14.9)$$

Inițial se consideră  $b^0 = N = (N_1, N_2, \dots, N_n)$  pentru care  $\Phi(N) = M$ , și  $b = 0 = (0, 0, \dots, 0)$ . Pentru descompunerea setului  $S$ , fiecărui set de stări acceptabile  $A$  i se asociază o valoare critică  $v^0 \in S$ , astfel încât:

$$A = \{X \in \mathcal{S}; v^0 \leq X\} \quad (3.14.10)$$

Fiecare set de stări neacceptabile  $B$  va avea o asociată de asemenea o valoare critică  $v_i$ , astfel încât:

$$B = \{X \in \mathcal{S}; x_i < v_i \text{ pentru cel puțin un } i\} \quad (3.14.11)$$

Fiabilitatea căutată este dată evident de probabilitatea de a avea o stare acceptabilă, adică:

$$R(k) = \Pr\{\Phi(X) \geq k\} = \Pr\{X \in A\} \quad (3.14.12)$$

Aceasta se poate calcula ușor ca o sumă de termeni, deoarece seturile de stări acceptabile generate conțin toate stările acceptabile și nu se suprapun (corespund la evenimente incompatibile).

Al doilea algoritm calculează în mod similar non-fiabilitatea la nivelul  $k$ , deci:

$$Q(k) = \Pr\{\Phi(X) < k\} \quad (3.14.13)$$

necesitând pentru aceasta cunoașterea *vectorilor de tăietură minimali* (minimal cut vectors). Un vector de stare  $Z$  formează o tăietură minimală dacă  $\Phi(Z) < k$ , dar  $\Phi(X) \geq k$  pentru orice  $X > Z$ . În [AVEN85] se arată superioritatea acestor algoritmi în ceea ce privește timpul de prelucrare pe calculator față de metodele de evaluare a fiabilității pe principiul incluziunii-excluziunii și al enumerării de stări.

Modelele de mai sus au presupus o ordonare totală a stărilor sistemului și a componentelor sale. O abordare mult mai flexibilă este aceea care împarte mulțimea vectorilor de stare ale componentelor pe clase de echivalență [BOED93] [BOED94] sau cea care consideră doar o ordonare parțială a acestor stări și duce astfel la un model generalizat al sistemului coerent cu stări multiple (generalized multistate coherent system) [YU94]. Acest model se poate aplica cu succes la acele sisteme practice care posedă mai mult de un singur parametru de performanță și care având cel puțin două moduri de degradare a performanțelor nu pot fi descrise prin mulțimi complet ordonate a stărilor.

## 4. FIABILITATEA SISTEMELOR MULTIPROCESOR

### 4.1. Caracteristici generale de fiabilitate și performanță

La calculul fiabilității unui sistem multiprocesor sunt necesare în principiu următoarele trei categorii de informații:

- a) fiabilitatea componentelor individuale ale sistemului
- b) structura fizică sau logică de interconectare a componentelor, care impune o caracteristică particulară unui sistem, având deci o influență majoră asupra fiabilității acestuia
- c) cerințele sau restricțiile impuse sistemului, care definesc condițiile limită ca sistemul să fie considerat operațional (de exemplu numărul minim de componente funcționale)

Metodele tradiționale de evaluare a fiabilității utilizează structura fizică de interconectare și cerințele de funcționalitate pentru a genera reprezentări intermediare de genul arborilor de defectare sau grafurilor de fiabilitate. Conceptele de fiabilitate se traduc în aceste cazuri în determinarea probabilității ca între oricare două noduri ale grafului să existe o cale de comunicație în orice moment sau posibilitatea ca să rămână suficiente arcuri operaționale pentru a menține cel puțin o ramură care să cuprindă toate nodurile.

La nivelul arborilor de defectare și al grafurilor pentru analize probabilistice se lucrează cu noduri și arce omogene, fără să se țină cont de diversitatea componentelor unui sistem. Pentru a elimina acest neajuns, s-a dezvoltat la universitatea Carnegie Mellon *modelul PMS* (processor-memory-switch) de reprezentare a unui sistem multiprocesor pentru calculul fiabilității [KINI82]. Acesta descrie structura sistemului la un nivel la care nodurile grafului de conexiune reflectă eterogenitatea componentelor. Criteriul de funcționalitate al sistemului este dat în acest caz de numărul minim de componente operaționale din fiecare tip și care sunt capabile de comunicare.

Un exemplu de structură PMS este prezentat în figura 4.1.1, unde două procesoare P1 și P2 pot comunica prin magistralele S1 și S2 cu memoriile M1, M2, M3 și cu unitățile de disc D1, D2. Cu  $K_m$  și  $K_d$  s-au notat controlerile de memorie, respectiv de disc. Este evident că o astfel de reprezentare este adecvată sistemelor strâns cuplate. Dacă se consideră că sistemul mai poate funcționa la limită cu un procesor, două memorii și un disc, aceste cerințe se pot formula printr-o declarație booleană de forma:

$$\Phi(1, P) \text{ \&I } \Phi(2, M) \text{ \&I } \Phi(1, D)$$

Kini și Siewiorek au pus la punct o metodă de determinare a funcției de fiabilitate sub formă simbolică pentru structuri arbitrare de tip PMS, algoritmul fiind implementat pe calculator. Ca intrări în program se specifică tipul și parametrii funcției de fiabilitate a componentelor (exponențială, Weibull, etc.), graful de interconectare PMS și declarația cerințelor minime de funcționalitate. Informații suplimentare referitoare la magistrale, elemente de comutare, etc. nu se dau în declarație, ele fiind deduse pe baza topologiei.

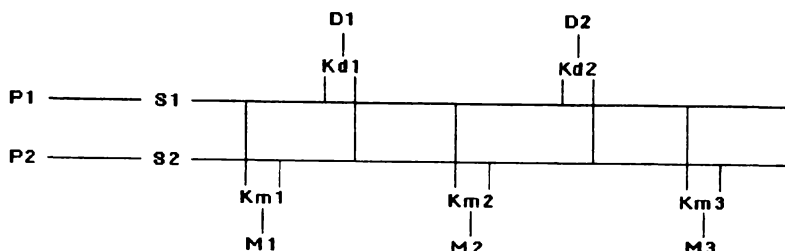


Figura 4.1.1.

Cerințele minimale se folosesc pentru generarea tuturor seturilor de resurse critice minimale, care conțin un număr de elemente vitale (procesoare, memorii, discuri) necesare pentru reușită. Garantarea funcționării componentelor dintr-un astfel de set încă nu este suficientă pentru starea de reușită, fiind necesară identificarea componentelor auxiliare care trebuie să fie funcționale. Acestea se vor găsi pe căile de comunicație dintre resursele vitale pe subgraful asociat fiecărui set. Astfel se obțin mulțimile de componente care dau stări de reușită, deci niște rezultate parțiale pentru care se calculează funcțiile de fiabilitate sub formă canonică cu regulile de calcul probabilistic. Forma canonică este necesară pentru a ușura reprezentarea codificată în binar a termenilor în calculator. Funcția de fiabilitate finală se calculează conform relației (3.7.1) ca probabilitatea disjuncției evenimentelor de reușită, considerate ca fiind evenimente independente.

O optimizare a programului s-a obținut prin segmentarea prealabilă a grafului PMS în subgrafuri pentru care există tehnici cunoscute de calcul a funcției de fiabilitate. De altfel metoda de determinare a fiabilității unui sistem bazată pe descompunerea în subrețele a fost prezentată la paragraful 3.6.

În investigarea sistemelor multiprocesor o atenție mărită este acordată fiabilității rețelei de conexiune dintre componente. Fiabilitatea globală a acestei rețele se definește ca probabilitatea de a menține capacitatea de acces complet prin rețea, iar fiabilitatea de tip terminal înseamnă probabilitatea ca să existe cel puțin o cale de comunicație între un anumit procesor și o anumită memorie (sau anumite memorii).

În general pentru evaluarea fiabilității unor astfel de sisteme multiprocesor se folosesc principii de *teoria grafurilor* îmbinate cu *analiza combinatorie*. Această metodă se preferă și în [HWANG82] cu referire la arhitecturile multiprocesor clasificate după rețeaua de interconectare în:

- structuri de tip magistrală, cu divizarea timpului (figura 4.1.2)
- structuri cu comutare crossbar (figura 4.1.3)
- structuri cu memorii multiport (figura 4.1.4)

Pe fiecare figură s-a reprezentat și graful probabilistic asociat, care este un set finit de noduri ponderate conectate prin muchii nedirecționate. Fiecare nod corespunde unei componente fizice, fiind ponderat cu fiabilitatea sa (probabilitatea de bună funcționare, considerat ca element independent, de forma  $x=e^{-\lambda t}$ ). De notat că în categoria elementelor de memorie sunt incluse atât memoriile principale, cât și dispozitivele periferice de stocare. Comunicarea între procesoare are loc prin intermediul memoriilor pe căi de comunicație DFP (data flow path) de tip procesor - element de conectare - memorie.

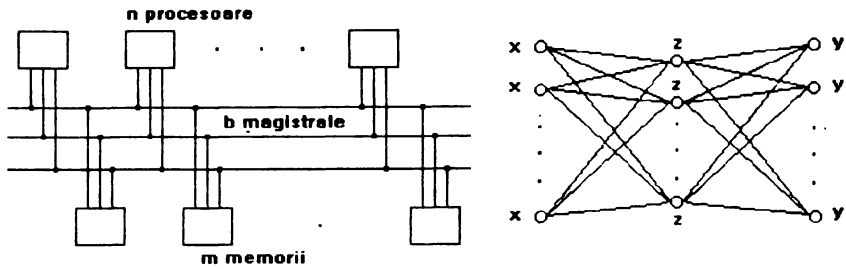


Figura 4.1.2.

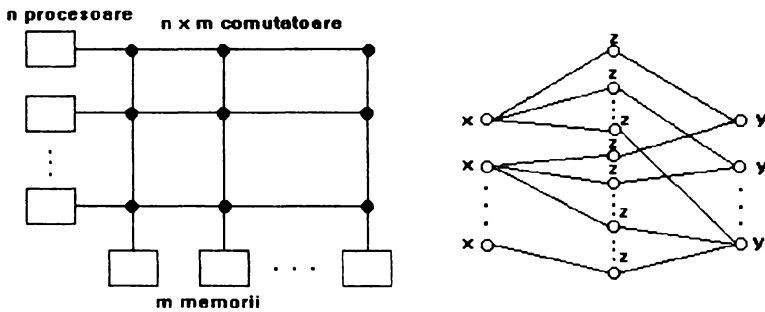


Figura 4.1.3.

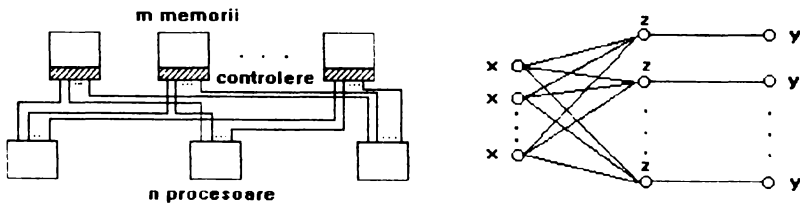


Figura 4.1.4.

De exemplu, pentru o structură de tip magistrală numărul maxim de căi este  $n \times b \times m$ , din care o parte trebuie să fie funcționale pentru ca sistemul să fie operațional. Se definește o *fiabilitate de prag* (threshold reliability):

$$R_{\alpha,\beta} = \Pr\{ E(\alpha,\beta) \}$$

care reprezintă probabilitatea de a avea cel puțin  $\alpha$  procesoare din  $n$  care să lucreze cu  $\beta$  memorii din  $m$ . Pentru calcularea acestei mărimi, în loc de a efectua numărarea pe graf a tuturor situațiilor favorabile, se recurge la calculul combinațional. Astfel, probabilitatea de

a avea exact  $i$  unități bune din  $n$ ,  $p$  fiind funcția de fiabilitate  $p = e^{-\lambda}$  a unității, este dată de relația:

$$G(n, i, p) = \binom{n}{i} p^i (1 - p)^{n-i} \tag{4.1.2}$$

Fiabilitatea de prag pentru structura considerată se obține simplu ca fiind:

$$R_{\alpha, \beta}(t) = \sum_{i=\alpha}^n \sum_{j=\beta}^m \sum_{k=1}^b G(n, i, x) G(m, j, y) G(b, k, z) \tag{4.1.3}$$

Fiabilitatea sistemului se poate calcula ca

$$R(t) = \Pr\{E(1,1)\} = R_{1,1}(t) \tag{4.1.4}$$

dar pentru ca sistemul să funcționeze în mod MIMD (multiple instruction multiple data) se consideră că sunt necesare cel puțin 2 procesoare și se definește o *fiabilitate de multiprocesare* (multiprocessing reliability):

$$R_m(t) = \Pr\{E(2,1)\} = R_{2,1}(t) \tag{4.1.5}$$

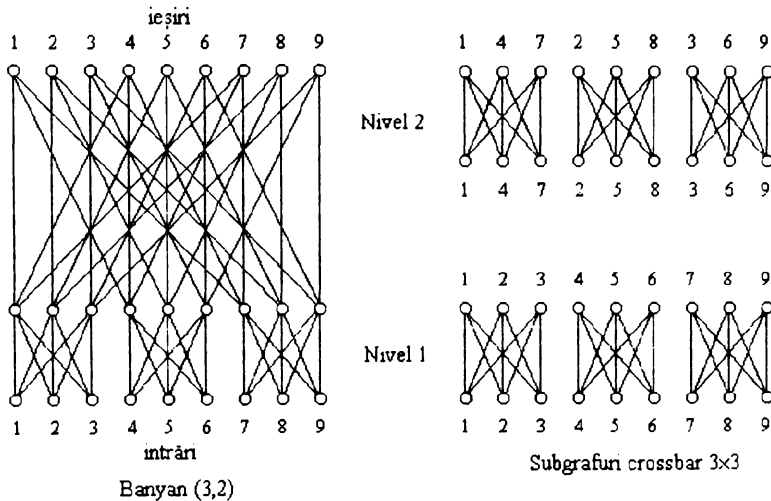


Figura 4.1.5.

Circuitele de tip crossbar pot fi întâlnite în componența acelor rețele de interconectare multinivel MIN la care există câte o singură cale între fiecare nod de intrare și nod de ieșire. O astfel de structură este *rețeaua banyan*. Un banyan de tip  $(f,L)$  este organizat pe  $L$  nivele, astfel încât factorul de intrare al fiecărui nod intermediar este egal cu factorul de ieșire, având valoarea  $f$  (fanout- $f$ ). Fiecare cale de la o intrare la o ieșire are

lungimea  $L$ . Un graf banyan  $(f, L)$  poate fi privit ca o colecție de  $f \times f$  subgrafuri de tip crossbar, așa cum se poate observa pe exemplul din figura 4.1.5. Aceasta reprezintă un MIN de dimensiune  $9 \times 9$  construit din 2 nivele de comutatoare crossbar  $3 \times 3$ , rețeaua fiind un banyan  $(3, 2)$ .

Fiabilitatea unei rețele banyan se poate calcula observând că ea constituie un sistem neredundant, un defect singular în rețea duce la deconectarea comunicării între anumite intrări și ieșiri [CHER85]. Într-o rețea banyan  $(f, L)$  rectangulară  $N = f^L$ , unde  $N$  reprezintă numărul de resurse (intrări/ieșiri), numărul de elemente de comutare necesare este evident  $(N/f) \times L$ . Considerând că elementele de comutare crossbar  $B_f$  se supun unei legi de fiabilitate exponențială, rata de defectare fiind  $\lambda(B_f)$ , rata de defectare a întregii rețele va fi:

$$\lambda_e = (N/f) \times L \times \lambda(B_f) \quad (4.1.6)$$

Rata de cădere  $\lambda(B_f)$  a unui element  $B_f$  este o funcție direct proporțională cu fanout-ul  $f$ , depinzând de implementarea schemei și de tehnologie, fiind calculabil de la caz la caz în funcție de acești factori.

Comportarea rețelei la defecte prin degradarea performanțelor se poate studia printr-un coeficient de slăbire la defect FS (fail-softness coefficient), exprimat ca raportul dintre numărul de conexiuni între intrări-ieșiri în caz de defect și numărul total de conexiuni posibile în lipsa defectărilor. Se deduce simplu că pentru un defect singular al unui nod

$$FS = (N^2 - N) / N^2 = 1 - 1/N \quad (4.1.7)$$

iar pentru un defect singular al unei legături:

$$FS = (N^2 - N/f) / N^2 = 1 - 1/(f \times N) \quad (4.1.8)$$

Întrucât rețeaua MIN constituie elementul esențial al sistemului multiprocesor, problemele de bază sunt cele de conectivitate și toleranță la defecte ale acestui circuit. Se consideră în [AGRA85] că sistemul este funcțional dacă el respectă proprietatea de *access dinamic complet* (DFA, dynamic full access). Aceasta înseamnă că fiecare intrare în MIN poate fi conectată la oricare din ieșiri după un număr finit de pași. Deficiența metodei este că se presupune existența defectelor (de punere pe "0" sau pe "1") numai la intrările de comandă și cu anumite rezerve pe cele de date ale elementelor de comutare. De altfel se demonstrează că un defect de agățare pe o intrare sau ieșire primară a MIN, duce la imposibilitatea asigurării proprietății DFA.

Testarea capacității DFA se face cu ajutorul modelării prin graf a circuitelor de comutare, ceea ce duce la un model de graf pentru întreaga rețea MIN. Figura 4.1.6 prezintă graful corespunzător rețelei MIN din figura 2.4.5. Pe baza acestui graf complet se pot scrie matricile de adiacență  $A$ , pentru fiecare nivel, care arată căile posibile între nodurile de intrare și de ieșire ale nivelului respectiv, și matricea de accesibilitate  $R$  (reachability matrix) pentru întregul graf, care indică căile posibile între intrările și ieșirile primare. Evident există relația:

$$R = \prod_{i=1}^m A_i \quad (4.1.9)$$



unde  $m$  reprezintă numărul de nivele. Matricea de accesibilitate în  $k$  pași se obține prin:

$$R_k = R^k \quad (4.1.10)$$

Condiția DFA este satisfăcută dacă se obține o matrice de accesibilitate (chiar în  $k$  pași) cu toate elementele nenule. Modelând pe graf diferite situații de defect, prin calcularea matricii de accesibilitate se poate face o afirmație asupra funcționalității sistemului. Simulând diverse modele de defect, se poate evalua pe baza numărului de pași o distanță medie de cale, care dă o indicație asupra performanțelor circuitului pentru structura de interconectare aleasă a elementelor de comutare. Invers, pe baza acestui principiu s-a conceput și o metodă optimă de sinteză a circuitelor MIN bazată pe o parționare pe grupuri, care duce la un număr de defecte tolerabile aproape egal cu numărul circuitelor de comutare. O optimizare în continuare trebuie căutată în direcția minimizării distanței medii.

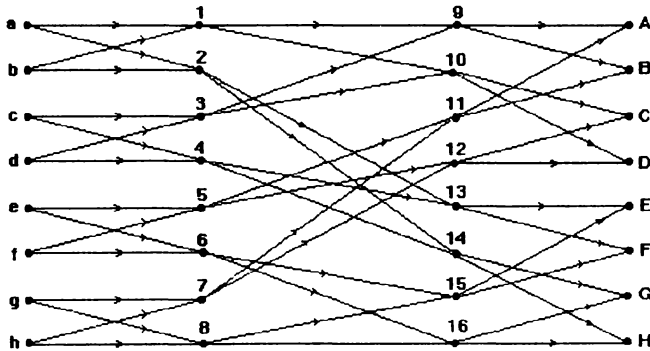


Figura 4.1.6.

Matricea de accesibilitate  $R$  este folosită în [MACA88] pentru a determina prin simulare gradul de degradare al sistemului în prezența unor defecte. Dacă un sistem strâns cuplat de tip MIN este definit ca fiind funcțional când cel puțin  $i$  procesoare și  $j$  module de memorie din  $N$  sunt operaționale și complet conectate între ele, atunci condiția de funcționalitate se poate determina din matricea  $R$ , și anume dacă în aceasta se găsește o submatrice de ordinul  $(i \times j)$  cu elemente nenule. Un astfel de model de fiabilitate se numește *fiabilitate orientată pe sarcini* (task-based reliability). Simulatorul reduce deci sistemul în prezența diferitelor defecte la matricea de accesibilitate, din care se determină o mărime a fiabilității.

Fiabilitatea unui sistem conținând  $N$  procesoare PE omogene, fără rezervă, și o rețea MIN de  $N \times N$  (chiar tolerantă), este dată de relația

$$R_0 = C_1 \times R_p^N \times R_n \quad (4.1.11)$$

unde:

$R_0$  - fiabilitatea sistemului fără procesoare de rezervă ( $S=0$ )

- $R_p$  - fiabilitatea unui singur PE
- $R_n$  - fiabilitatea rețelei MIN
- $C_1$  - coeficient de degradare a fiabilității din cauza unor elemente secundare (circuite de comandă, memorii secundare)

Mărimea lui  $R_0$  are o limită superioară  $R = C_1 \times R_p^N$  care nu poate fi depășită nici dacă rețeaua este perfectă ( $R_n = 1$ ). În schimb, utilizând un procesor de rezervă se poate trece de această valoare, fiabilitatea întregului sistem, cu redundanță dinamică DR (paragraful 2.4), fiind dată în acest caz de

$$R_1 = C_2 [(R_p R_r)^{N+1} + (N+1) (R_p R_r)^N (1-R_p R_r)] \quad (4.1.12)$$

unde:

- $R_1$  - fiabilitatea sistemului cu  $S=1$
- $R_r$  - fiabilitatea unui rând al rețelei DR
- $C_2$  - coeficient similar lui  $C_1$

În [JENG88] se demonstrează pe exemple că pentru  $N$  mic rezultă  $R_1 > R$ , iar pentru  $N$  foarte mare  $R_1$  poate deveni mai mic decât  $R$ , deoarece adăugarea unei rezerve nu compensează scăderea de fiabilitate datorată dimensiunii mari a rețelei. Totuși se va menține  $R_1 > R_0$ , întrucât fiabilitatea unui MIN tolerant la defecte se va micșora și ea odată cu creșterea lui  $N$  ( $R_n$  scade). În concluzie, deși mai scumpă decât soluția cu rețea MIN cubică, rețeaua cu redundanță dinamică DR duce la creșterea semnificativă a fiabilității sistemului pentru un număr mare de procesoare.

Defectele dintr-o rețea MIN cu înlănțuire (figura 2.4.9) pot fi reunite pe elemente, un element fiind constituit din ieșirea (sau ieșirea-lanț) a unui element de comutare SE, legătura aferentă și intrarea (sau intrarea-lanț) la care merge în comutatorul SE următor. Un element se consideră defect dacă oricare componentă a sa este defectă. Defectele de pe intrările și ieșirile primare ale rețelei nu pot fi tolerate decât prin utilizarea unor comutatoare externe suplimentare [TZENG88].

Pentru analiza fiabilității trebuie mai întâi calculat numărul minim de elemente lipsite de defect necesare pentru a asigura conectivitatea completă a rețelei. Un lanț complet de  $M/2$  circuite SE are  $M$  elemente de ieșire și  $M/2$  elemente de ieșire-lanț. Cazul cel mai defavorabil pe care un lanț îl poate tolera este când toate elementele de ieșire ale lanțului complet sunt defecte în afară de două (dar nu oricare, pentru a se asigura căte o cale către ambele grupări din nivelul următor) și toate elementele de ieșire-lanț sunt bune exceptând unul singur. Deci un lanț complet poate tolera cel mult  $(M-2) + 1 = M-1$  elemente defecte. Un nivel  $i$  având  $2^i$  lanțuri complete de câte  $N/2^{i+1}$  circuite SE și fiindcă  $0 \leq i \leq \log_2 N-1$ , numărul maxim de defecte care pot fi tolerate este:

$$B = \sum_{i=0}^{\log_2 N-2} 2^i \left( \frac{N}{2^i} - 1 \right) = N \left( \log_2 N - \frac{3}{2} \right) + 1 \quad (4.1.13)$$

Pentru cazul din figura 2.4.9 (cu  $N=8$ ) se obține  $B=13$ . Este evident că defectarea unui singur element din rețea este totdeauna tolerată, dar  $B+1$  defecte vor duce sigur la căderea rețelei. Între aceste două extreme se situează numărul de defecte tolerabile, în funcție de locurile de apariție ale defecțiunilor.

Funcția de fiabilitate a rețelei se poate calcula cu relația:

$$R(t) = Q(0)R(0) + Q(1)R(1) + \dots + Q(E)R(E) \quad (4.1.14)$$

unde  $Q(i)$  reprezintă probabilitatea de supraviețuire a rețelei după căderea a  $i$  elemente, iar  $R(i)$  este fiabilitatea rețelei cu  $i$  elemente defecte. Numărul de elemente expuse defectării, fără a admite defecte pe intrările și ieșirile primare, este  $E = 3N(\log_2 N - 1)/2$ . Calculul probabilităților  $Q(i)$  se poate face printr-o metodă recurentă în ordinea descrescătoare a nivelelor [TZENG88], cu observația că  $Q(i) = 0$  pentru  $i > B$ . Admițând pentru fiecare element o distribuție exponențială a timpilor de defectare cu rata  $\lambda$ , probabilitatea de a avea  $i$  elemente defecte din  $E$  este:

$$R(i) = \binom{E}{i} (1 - e^{-\lambda t})^i (e^{-\lambda t})^{E-i} \quad (4.1.15)$$

Înlocuind mai sus și având în vedere observația făcută, rezultă:

$$\begin{aligned} R(t) &= \sum_{i=0}^B Q(i) \binom{E}{i} (1 - e^{-\lambda t})^i (e^{-\lambda t})^{E-i} = \\ &= (e^{-\lambda t})^{E-1} + \sum_{i=1}^B Q(i) \binom{E}{i} (1 - e^{-\lambda t})^i (e^{-\lambda t})^{E-i} \end{aligned} \quad (4.1.16)$$

Media timpului de funcționare se poate calcula ca:

$$MTTF = \int_0^{\infty} R(t) dt = \frac{1}{\lambda} \sum_{i=0}^B \frac{Q(i)}{E-i} \quad (4.1.17)$$

Pentru comparație, funcția de fiabilitate a unei rețele regulate neredundante, cu același număr de intrări/ieșiri  $N$ , este:

$$R_0(t) = (e^{-\lambda t})^{E_0} \quad (4.1.18)$$

unde  $E_0 = N(\log_2 N - 1)$ , admițând și aici intrările și ieșirile primare perfect fiabile. În acest caz  $MTTF_0$  este evident:

$$MTTF_0 = \frac{1}{\lambda E_0} \quad (4.1.19)$$

Reprezentând grafic (la scară logaritmică)  $MTTF$  în funcție de  $N$  pentru rețeaua înlănțuită cu redundanță, respectiv pentru cea fără redundanță, se obține graficul din figura 4.1.7 [KAKAS94a]. Se observă că odată cu creșterea dimensiunii rețelei, fiabilitatea rețelei redundante are o tendință mai mică de scădere față de fiabilitatea rețelei regulate. Acest lucru se explică prin faptul că se obțin lanțuri din ce în ce mai mari, capabile să furnizeze mai multe căi alternative, al căror număr crește exponențial cu dimensiunea rețelei. Cu cât rețeaua este mai mare ( $N$  mare), cu atât mai bine poate

supraviețui o astfel de rețea complet înlănțuită la un număr mai mare de defecte, deci devine mai robustă. În ceea ce privește costul rețelei înlănțuite, aceasta este cu aproximație de 9/4 ori mai mare decât cel al rețelei regulate, deoarece circuitele de comutare de tip  $2 \times 2$  trebuie înlocuite cu circuite de tip  $3 \times 3$ .

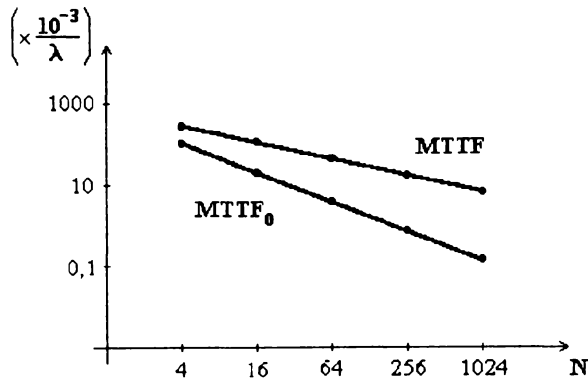


Figura 4.1.7.

**Indicatori de performanță.** Scopul rețelei de interconectare a unui sistem multiprocesor este de a face legătura între oricare procesor și un modul de memorie din zona de memorie comună. Satisfacerea unei astfel de cereri poate fi însă împiedicată din cauza a două motive principale:

1. Dacă două sau mai multe procesoare cer acces simultan la aceeași unitate de memorie.
2. Dacă două sau mai multe procesoare necesită simultan aceeași legătură de comunicație pentru a accesa unități de memorie distincte.

În ambele cazuri procesoarele care au emis cereri de acces nereușite trebuie să aștepte. Structura rețelei de interconectare trebuie să permită menținerea la minim a timpului pierdut de procesoarele care așteaptă din cauza memoriilor sau resurselor de comunicație ocupate. Eliminarea pe cât posibil a oricărei dispute pentru resurse comune ar necesita structuri de interconectare foarte complexe și costisitoare, dar sistemele actuale (prezentate în capitolul 2) prezintă un compromis între complexitate și cost care nu reduce în mod semnificativ performanțele sistemului multiprocesor oferite de prelucrarea paralelă. La acest lucru contribuie și faptul că memoria totală este de regulă partiționată într-o memorie locală (compusă din module independente asociate fiecărui procesor) și o memorie comună (compusă din module accesibile de către oricare procesor). Ca urmare, un procesor se poate afla într-una din următoarele stări [AJMO86]:

- Activ (active): procesorul lucrează cu memoria sa locală.
- În accesare (accessing): procesorul schimbă date cu alte procesoare prin înscrierea în (sau citirea din) zona de memorie comună.
- În coada de așteptare (queued): procesorul așteaptă obținerea unui acces la memoria comună.

O parte importantă a modelului de evaluare a performanțelor constă în caracterizarea încărcării (workload) aplicate sistemului. În acest sens există două abordări posibile: deterministă și probabilistă. În cazul *determinist* încărcarea fiecărui procesor se consideră a fi cunoscută și pe această bază se poate dezvolta o secvență precisă a operațiilor efectuate de fiecare procesor (durata activităților singulare, destinate fiecărei cereri de acces) pentru a caracteriza comportarea întregului sistem. Parametrii de performanță se pot evalua prin simulare, ceea ce limitează această tehnică la cazul concret aflat sub observare. Admițând în schimb o comportare *probabilistă* a sistemului, se poate dezvolta un model stohastic care încearcă să capteze esențialul în funcționarea sistemului, fără a fi cunoscute detaliile individuale ale încărcării sistemului. Parametrii de încărcare se definesc ca și variabile aleatoare cu o anumită distribuție, rezultatele furnizate având avantajul de a reprezenta relațiile funcționale între parametrii de sistem și performanță. De asemenea, un astfel de model stohastic permite studiul comparativ al diferitelor soluții arhitecturale cu referire la obiectivele de performanță impuse.

Unele modele de performanță admit o funcționare *asincronă* a procesoarelor corespunzătoare execuției segmentelor de program și acceselor la memoria comună pentru un transfer de date. Activitatea lor poate fi privită ca și secvențe de accese asincrone la diferite module de memorie (locale sau comune), apărând coliziuni dacă mai multe procesoare au nevoie de aceleași resurse. Abordările de tip *sincron* consideră sistemul la nivelul instrucțiunilor, reprezentate ca operații sincrone pentru toate procesoarele. Axa timpului este divizată pe intervale elementare numite "slot"-uri, astfel încât la începutul fiecărui slot procesoarele pot emite cereri de acces la memoria comună (figura 4.1.8). Apoi, mecanismul de arbitraj selectează cererile care pot fi onorate și procesoarele se conectează la modulele de memorie, conexiunile fiind menținute pe toată durata slotului. Timpul de lucru al fiecărui element de procesare (PE, processing element) este împărțit în:

- timp activ ( $t_a$ ), în care procesorul lucrează singur, eventual cu memoria locală
- timp de lucru cu memoria ( $t_m$ ), în care necesită acces la un modul de memorie comună CM (common memory)

Ciclu de execuție al procesorului se studiază sub aspectul raportului dintre timpul activ al procesorului (processor active time) și timpul de citire-scriere în memoria comună (memory write time). Dacă se notează cu  $\alpha$  probabilitatea ca în slotul care începe PE să emită o cerere de acces la CM, atunci pentru  $\alpha = 1$  timpul activ efectiv al procesorului va fi nul, iar pentru  $\alpha < 1$  timpul activ va avea o distribuție geometrică.

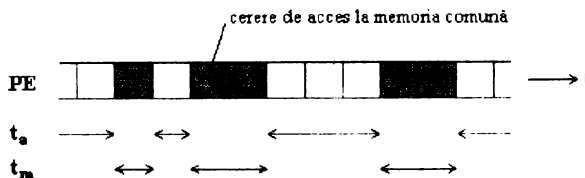


Figura 4.1.8.

Performanța sistemului multiprocesor este caracterizată cel mai adesea prin lățimea de bandă (bandwidth), care reprezintă numărul mediu de module de memorie

comună ocupate. O altă mărime pentru studiul degradării performanțelor datorită interferențelor de acces la resurse comune constă în numărul mediu de procesoare aflate în lucru, numită și *puterea de procesare* (processing power). În continuare voi face o trecere în revistă a expresiilor pentru calculul lățimii de bandă ( $\beta$ ) pe baza [WILK91] și [AJMO86].

Probabilitatea ca un PE să nu ceară un acces la CM la începutul unui slot este  $1-\alpha$ . Dacă  $n$  reprezintă numărul de procesoare, atunci probabilitatea ca nici un PE să nu facă cerere este  $(1-\alpha)^n$ , respectiv probabilitatea ca unul sau mai multe PE să facă cerere este  $1-(1-\alpha)^n$ . Deoarece într-un sistem cu *magistrală unică* numai o singură cerere poate fi acceptată la un moment dat, numărul mediu de cereri acceptate în fiecare ciclu de arbitrare, adică lățimea de bandă este dată de:

$$\beta = 1 - (1 - \alpha)^n \quad (4.1.20)$$

Trebuie făcută observația că în cazul acestui tip de sistem multiprocesor, când există o singură memorie comună accesibilă prin magistrala sistem, lățimea de bandă exprimă de fapt gradul de ocupare al acestui modul de memorie. Este evident că o rată ridicată a cererilor ( $\alpha$  mare) duce la saturarea rapidă a magistralei în funcție de numărul de procesoare.

În cazul sistemelor *crossbar*, coliziunile apar numai din cauza magistralelor memoriilor comune, nu și a magistralelor procesoarelor, deoarece fiecare procesor dispune de propria magistrală, dar mai multe procesoare pot concura pentru un modul de memorie și magistrala aferentă. Admițând șanse egale ca un procesor să poată accesa orice modul de memorie comună, probabilitatea ca procesorul PE<sub>*i*</sub> să ceară un acces la memoria CM<sub>*j*</sub> este  $\alpha/m$ , unde  $m$  este numărul de module de memorie comună. Probabilitatea ca PE<sub>*i*</sub> să nu ceară acces la CM<sub>*j*</sub> este  $1-\alpha/m$ , iar probabilitatea ca nici un procesor din cele  $n$  să nu ceară accesul la această memorie este  $(1-\alpha/m)^n$ . Probabilitatea ca modulul de memorie considerat să aibă cel puțin o cerere va fi  $1-(1-\alpha/m)^n$ . Ca urmare numărul de module de memorie comună care vor avea cel puțin o cerere, adică lățimea de bandă a sistemului este dată de:

$$\beta = n \left[ 1 - \left( 1 - \frac{\alpha}{m} \right)^n \right] \quad (4.1.21)$$

Performanțele pot fi limitate în cazul sistemelor cu *magistrale multiple* atât de disputele pentru magistrale, cât și pentru memoriile comune. Într-un ciclu de magistrală pot fi servite maximum  $b$  cereri ( $b$  fiind numărul de magistrale), iar apoi cele  $b$  cereri trebuie să fie către module de memorie diferite. Servirea unei cereri de acces poate fi deci privită ca și un proces în două etape. Prima dată trebuie selectate până la  $m$  cereri dintre cele emise. Probabilitatea ca o memorie să aibă cel puțin o cerere este, ca și la rețeaua *crossbar*,  $1-(1-\alpha/m)^n = r$ . Urmează că probabilitatea ca exact  $i$  module de memorie diferite să aibă cereri adresate este, conform unei distribuții binomiale:

$$g(i) = \binom{m}{i} r^i (1-r)^{m-i} \quad (4.1.22)$$

În a doua etapă, din toate cererile diferite remise, numai  $b$  pot fi servite, datorită numărului limitat de magistrale. Rezultă că lățimea de bandă totală este dată de:

$$\beta = \sum_{i=b}^m bg(i) + \sum_{i=1}^{b-1} ig(i) \quad (4.1.23)$$

Primul termen se referă la situația când s-au efectuat  $b$  sau mai multe cereri pentru care s-au folosit toate magistralele, iar celălalt termen se referă la situația când au existat mai puțin de  $b$  cereri, necesitând evident și un număr egal de magistrale. Pentru situația  $m < b$ , primul termen dispare. (Relațiile (4.1.22) și (4.1.23) constituie și o corecție față de [WILK91], unde sunt date greșit, cu  $n$  în loc de  $m$ . În plus, voi arăta la paragraful 4.2 că aceste relații sunt valabile numai pentru  $n \geq m$ , pentru  $n \leq m$  modalitatea de calcul fiind alta). În [MELUS88] se modelează un sistem cu două nivele de magistrale, lățimea de bandă totală având trei componente, corespunzătoare accesului local, accesului în interiorul grupului propriu și accesului la o memorie dintr-un alt grup (cluster).

Probabilitatea ca cererea emisă de un procesor să fie acceptată, este dată la toate sistemele parcurse până acum de raportul dintre numărul de cereri acceptate și numărul de cereri emise, adică:

$$P_A = \frac{\beta}{n\alpha} \quad (4.1.24)$$

La un sistem cu rețea multinivel care are rețeaua compusă din comutatoare crossbar de tip  $\alpha \times b$  în fiecare nivel, numărul de cereri acceptate de un astfel de comutator aflat în primul nivel și transmis spre cel de-al doilea, este, conform (4.1.21),  $b[1 - (1 - \alpha/b)^2]$ . Probabilitatea ca să existe o cerere reușită pe oricare nivel din cele  $b$  ieșiri ale primului nivel, este:

$$r_1 = 1 - \left(1 - \frac{\alpha}{b}\right)^2 \quad (4.1.25)$$

Aceste cereri devin intrări în al doilea nivel, și așa mai departe, astfel încât probabilitatea ca să existe o cerere reușită pe o ieșire a ultimului nivel, se poate afla prin evaluarea recurentă a funcției:

$$r_i = 1 - \left(1 - \frac{r_{i-1}}{b}\right)^2 \quad (4.1.26)$$

pentru  $i = 1, 2, \dots, N$ , unde  $N$  este numărul de nivele și  $r_0 = \alpha$ . Întrucât rețeaua are în total  $b^N$  ieșiri (și  $\alpha^N$  intrări), lățimea de bandă totală a sistemului este:

$$\beta = b^N r_N \quad (4.1.27)$$

Probabilitatea ca o cerere să fie acceptată este dată în acest caz de:

$$P_A = \frac{b^N r_N}{a^N \alpha} \quad (4.1.28)$$

Relațiile precedente se pot particulariza pentru cazul rețelelor MIN cu celule de comutare  $2 \times 2$ , luând  $a = b = 2$ .

O modelare mai apropiată de realitate se obține dacă rata de acces statică  $\alpha$  se înlocuiește cu o rată de cereri dinamică (dynamic request rate)  $\delta$ , care ține cont de remiterea cererilor neservite de acces la memoria comună în slotul următor. Figura 4.1.9 prezintă timpul de lucru  $T$  al unui procesor cu toate cererile acceptate și timpul  $T'$ , în care unele cereri au fost blocate și reluate în ciclul următor. Deoarece numărul de sloturi fără cereri este același în ambele cazuri, există egalitatea:

$$T(1 - \alpha) = T'(1 - \delta) \quad (4.1.29)$$

Dacă  $P_A$  reprezintă probabilitatea ca o cerere să fie acceptată (calculată cu rata  $\delta$ ), atunci numărul așteptat de sloturi până la acceptare (inclusiv acesta) este media unei distribuții geometrice, și anume  $1/P_A$ . Timpul  $T'$  va fi deci mai lung decât  $T$  cu timpul corespunzător refuzurilor, adică:

$$T' = T + \alpha T \left( \frac{1}{P_A} - 1 \right) = \left[ (1 - \alpha) + \frac{\alpha}{P_A} \right] T \quad (4.1.30)$$

Din (4.1.28) și (4.1.30) se obține rata dinamică ( $\delta > \alpha$ ):

$$\delta = \frac{1}{1 + P_A \left( \frac{1}{\alpha} - 1 \right)} \quad (4.1.31)$$

Întrucât  $P_A$  este tot o funcție de  $\delta$ , având expresia (4.1.24) sau (4.1.28) în care se înlocuiește  $\alpha$  cu  $\delta$ , rezolvarea se poate face prin iterații succesive pornind cu  $\delta = \alpha$ .

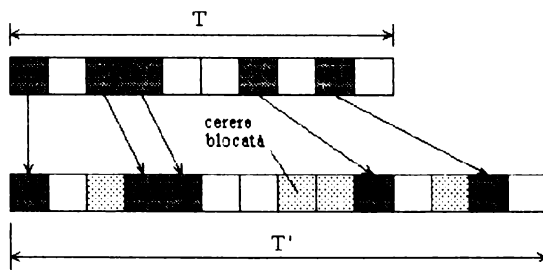


Figura 4.1.9.

Un alt indicator de performanță al sistemelor multiprocesor este *factorul de accelerare* (speed-up factor). Dacă toate cererile ar fi acceptate, timpul de execuție pe sistemul multiprocesor ar fi  $T$ , și factorul de accelerare  $n$ , având în vedere că pe sistemul



monoprocesor timpul necesar ar fi  $nT$ . În cazul în care există și cereri refuzate datorită accesului simultan la resurse comune, factorul de accelerare este:

$$F_a = \frac{nT}{T'} = \frac{n}{(1 - \alpha) + \frac{\alpha}{P_A}} \quad (4.1.32)$$

Chiar cu rata dinamică a cererilor, față de comportarea reală modelele considerate sunt aproximative, deoarece am admis șanse egale pentru toate procesoarele de a avea un acces la memoria comună. Modele mai exacte se obțin decă se ține cont de tehnica de arbitrarie și de comportarea unor anumite procesoare de a selecta cu preferință anumite module de memorie comună [AJMO86]. De asemenea, simularea pe calculator poate fi folosită pentru a obține rezultate mai precise.

#### 4.2. Modelul integrat de fiabilitate și performanță propus

Studiul performanțelor sistemelor multiprocesor se face de regulă ținând cont de caracteristicile lor constructive și funcționale. O analiză mai realistă se obține dacă se integrează indicatorii de performanță ai sistemului și mărimile de fiabilitate. Voi construi prima dată un astfel de model pentru un sistem de tip crossbar (prezentat și în [KAKAS95d]), urmând ca pe baza lui să fac referire și la alte tipuri de sisteme.

Consider un sistem multiprocesor de tip crossbar cu  $n$  elemente procesoare (PE),  $m$  module de memorie comună (CM) și  $n \times m$  comutatoare (figura 2.4.1). Elementele procesoare pot lucra independent, având acces la câte o memorie locală (LM) sau pot cere acces la memoria comună pentru un transfer de date. Dacă sosesc cereri simultane de la mai multe procesoare la același modul de memorie, atunci va fi satisfăcută una singură, urmând ca procesoarele neservite să intre în așteptare. Se va considera că elementele de comutare sunt expuse la defectare. Un comutator defect întrerupe calea dintre un PE și CM și poate cauza de asemenea plasarea în așteptare a procesorului, dacă a emis o cerere la acea memorie comună pe a cărei cale se află comutatorul. Astfel, un procesor poate să se afle într-una din următoarele stări:

- Activ (active), când lucrează cu memoria locală sau execută operații interne
- Ocupat (busy), când schimbă date cu o memorie comună
- În așteptare (waiting), când o cerere către memoria comună nu a fost servită

Pentru a modela comportarea sistemului prezentat mai sus, fac următoarele presupuneri [KAKAS95c]:

- Axa timpului se împarte pentru fiecare procesor pe intervale elementare (slot-uri), deci consider o evoluție discretă a timpului (figura 4.2.1). Un slot poate corespunde unei instrucțiuni de program sau unui ciclu mașină.

- Sistemul are o funcționare sincronă, adică ciclurile de procesoare și de memorie au aceeași durată și încep simultan.

- Fiecărui procesor  $i$  se asociază o probabilitate  $\alpha$  ca la slotul care urmează să efectueze o cerere de acces la memoria comună. Alegerea modulului de memorie la care se cere accesul se face după o distribuție uniformă. Modelul este pentru  $\alpha \neq 1$  un model cu întârziere între cereri (delay model), iar pentru  $\alpha = 1$  un model fără întârziere (zero delay model), adică fără procesare locală.

- Procesoarele, respectiv modulele de memorie comună sunt identice și nu se defectează.

- Comutatoarele sunt identice și fiecărui comutator  $i$  se asociază o probabilitate  $q=1-p$  de a se defecta în slotul care urmează ( $p$  este probabilitatea de a fi operațional).

- Dacă la o memorie sosesc simultan mai multe cereri, se reține una singură (aleasă în mod aleator), iar restul se pierde. Acesta este un model aproximativ cu cereri pierdute (lost requests), urmând ca mai încolo să aduc ajustările de rigoare.

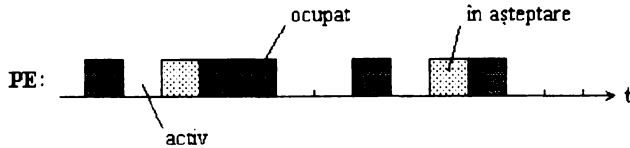


Figura 4.2.1.

Probabilitatea ca la un anumit slot să existe o cerere de acces reușită la un CM din partea unui PE, este dată de:

$$P = \alpha \cdot P_A \quad (4.2.1)$$

unde  $\alpha$  este probabilitatea ca să existe o cerere de acces, iar  $P_A$  reprezintă probabilitatea ca cererea să fie reușită. Aceasta din urmă se poate calcula ca:

$$P_A = P_B \cdot P_C \quad (4.2.2)$$

unde  $P_B$  exprimă probabilitatea ca drumul de legătură dintre PE și CM să fie operațional, iar  $P_C$  probabilitatea ca cererea să fie acceptată de modulul de memorie. Urmează determinarea lui  $P_B$  și  $P_C$ .

Pentru a determina pe  $P_B$ , trebuie să avem în vedere că probabilitatea de defectare a legăturii (de fapt a comutatorului) la fiecare slot este  $q$ . Astfel, fără a admite posibilitatea reparației,  $P_B$  reprezintă probabilitatea ca până la slotul considerat (al  $t$ -lea) să nu fi apărut defecțiunea ( $T$ ), adică este o funcție de fiabilitate dată de:

$$P_B = \Pr\{ T > t \} \quad (4.2.3)$$

Aici intervine în calcul mărimea timp, reprezentată prin variabila discretă  $t$ , adică numărul de slot-uri ( $t = 1, 2, 3, \dots$ ) de la momentul inițial al observării sistemului, de exemplu de la începutul rulării programului, când toate comutatoarele se consideră a fi bune. Deoarece timpul nu este continuu, în locul distribuției exponențiale se poate folosi aici distribuția de tip discret fără memorie, și anume distribuția geometrică. Aceasta are funcția de masă de probabilitate  $f(t) = q(1-t)^{t-1}$  și atunci:

$$\begin{aligned} P_B &= \Pr\{ T > t \} = f(t+1) + f(t+2) + f(t+3) + \dots = \\ &= q p^t + q p^{t+1} + q p^{t+2} + \dots = \\ &= q p^t (1 + p + p^2 + \dots) = \\ &= q p^t / (1 - p) = p^t \end{aligned} \quad (4.2.4)$$

Aceasta este de fapt probabilitatea ca pe durata a  $t$  slot-uri să nu apară defectarea unui anumit comutator.

Deoarece sunt posibile cereri simultane la un CM, probabilitatea ca cererea să fie acceptată de modulul de memorie ( $P_C$ ) se poate scrie ca:

$$\begin{aligned} P_C &= 1 \cdot \Pr\{\text{să existe doar această cerere care ajunge la modulul CM}\} + \\ &+ (1/2) \cdot \Pr\{\text{să existe încă 1 cerere care ajunge la modulul CM}\} + \dots + \\ &+ (1/n) \cdot \Pr\{\text{să existe încă } n-1 \text{ cereri care ajung la modulul CM}\} = \\ &= \sum_{k=0}^{n-1} \frac{1}{k+1} P_k \end{aligned} \quad (4.2.5)$$

Dacă se notează  $P_k = \Pr\{\text{să existe încă } k \text{ cereri care ajung la modulul CM}\}$ , atunci aceasta se poate calcula ca și suma următoarelor probabilități corespunzătoare la evenimente disjuncte:

$$\begin{aligned} P_k &= \Pr\{\text{să existe încă } k \text{ cereri lansate către acest CM și cele } k \text{ căi sunt bune}\} + \\ &+ \Pr\{\text{să existe încă } k+1 \text{ cereri lansate către acest CM, dar 1 cale să fie defectă} \\ &\text{din cele } k+1\} + \dots + \Pr\{\text{să existe încă } n-1 \text{ cereri lansate către acest CM,} \\ &\text{dar } n-k \text{ căi să fie defecte din cele } n-1\} = \sum_{i=0}^{n-k-1} P_{k,i} \end{aligned} \quad (4.2.6)$$

Evaluarea lui  $P_{k,i} = \Pr\{\text{să existe încă } k+i \text{ cereri lansate către acest CM, dar } i \text{ căi să fie defecte din cele } k+i\}$  se poate face ținând cont de independența evenimentelor:

$$\begin{aligned} P_{k,i} &= \Pr\{\text{să existe încă } k+i \text{ cereri către acest CM}\} \cdot \\ &\cdot \Pr\{k \text{ căi din cele } k+i \text{ să fie bune}\} \end{aligned} \quad (4.2.7)$$

Pentru a calcula primul factor al relației (4.2.7), este necesar mai întâi să se cunoască probabilitatea  $p_m$  ca la un slot să existe o cerere de acces din partea unui anumit procesor la un anumit modul CM. Admițând distribuția uniformă a cererilor, rezultă:

$$p_m = \frac{\alpha}{m} \quad (4.2.8)$$

Acum este evident că din punctul de vedere al unui anumit modul CM (oricare, ele fiind identice ca și comportare),  $\Pr\{\text{să existe încă } k+i \text{ cereri către acest CM}\}$  reprezintă probabilitatea ca din  $n-1$  încercări (fiecare cu probabilitatea de reușită  $\alpha/m$ ),  $k+i$  să fie reușite. Deci este cazul unei distribuții binomiale, astfel încât:

$$\begin{aligned} &\Pr\{\text{să existe încă } k+i \text{ cereri către acest CM}\} = \\ &= \binom{n-1}{k+i} p_m^{k+i} (1-p_m)^{n-1-(k+i)} = \binom{n-1}{k+i} \left(\frac{\alpha}{m}\right)^{k+i} \left(1-\frac{\alpha}{m}\right)^{n-1-(k+i)} \end{aligned} \quad (4.2.9)$$

Al doilea factor al relației (4.2.7) este dat simplu de reuniunea tuturor situațiilor în care  $k$  căi sunt bune și  $i$  defecte:

$$\Pr\{k \text{ căi din cele } k+i \text{ să fie bune}\} = \\ = \binom{k+i}{k} P_B^k (1-P_B)^i = \binom{k+i}{k} p^{ik} (1-p^i)^i \quad (4.2.10)$$

Înlocuind pe (4.2.9) și (4.2.10) în (4.2.7) și apoi mai departe până la (4.2.2), se obține probabilitatea ca cererea de acces la CM din partea unui PE la momentul  $t$  să fie reușită:

$$P_A = P_B \sum_{k=0}^{n-1} \sum_{i=0}^{n-k-1} \frac{1}{k+1} P_{k,i} = \\ = p^t \sum_{k=0}^{n-1} \sum_{i=0}^{n-k-1} \frac{1}{k+1} \binom{n-1}{k+i} \left(\frac{\alpha}{m}\right)^{k+i} \left(1-\frac{\alpha}{m}\right)^{n-1-(k+i)} \binom{k+i}{k} p^{ik} (1-p^i)^i \quad (4.2.11)$$

Conform relației (4.2.1) se poate calcula imediat și  $P$ :

$$P = \alpha p^t \sum_{k=0}^{n-1} \sum_{i=0}^{n-k-1} \frac{1}{k+1} \binom{n-1}{k+i} \left(\frac{\alpha}{m}\right)^{k+i} \left(1-\frac{\alpha}{m}\right)^{n-1-(k+i)} \binom{k+i}{k} p^{ik} (1-p^i)^i \quad (4.2.12)$$

Ca și un caz particular pentru  $\alpha=1$ , relațiile (4.2.11) și (4.2.12) sunt de asemenea valabile pentru un model fără întârziere cu cereri pierdute. Nici acest model nu ține însă cont de rata reală de acces la memoriile comune, care este  $\delta > \alpha$ , presupunând că o cerere neservită se va repeta în slotul următor. Un model îmbunătățit se poate obține dacă în (4.2.11) se consideră o rată dinamică a cererilor ( $\delta$ ) în loc de cea statică ( $\alpha$ ), relația de calcul fiind deja dedusă și dată de (4.1.31).

Ecuatiile (4.1.31) și (4.2.11), în care  $\alpha$  s-a înlocuit cu  $\delta$  permit o rezolvare recurentă, pornind cu  $\delta = \alpha$ . Rezultatele obținute pe calculator pentru  $l$  iterații până când  $|\delta^{(l)} - \delta^{(l-1)}| < 0,001$ , sunt date în tabelul 4.2.2 pentru diferite configurații de sistem și  $p=1$ . Este de remarcat faptul că eroarea introdusă de modelul aproximativ crește dacă numărul de module de memorie comună este prea mic comparativ cu numărul de procesoare și rata de cereri este ridicată.

$n$	$m$	$\alpha$	$P_A(\alpha)$	$\delta$	$P_A(\delta)$	$l$	$\Delta P_A / P_A$ [%]
10	6	0,3	0,803	0,357	0,770	4	3,9
		0,5	0,697	0,606	0,649	4	6,3
		0,7	0,609	0,804	0,569	4	6,6
5	5	0,3	0,887	0,328	0,877	3	1,1
		0,5	0,819	0,555	0,801	3	2,1
		0,7	0,757	0,759	0,739	3	2,3
7	9	0,3	0,905	0,323	0,899	3	0,7
		0,5	0,848	0,544	0,836	3	1,4
		0,7	0,795	0,749	0,782	3	1,5

Tabelul 4.2.2.

În ceea ce privește *durata de viață utilă* a sistemului, aceasta poate fi definită în mai multe feluri, în funcție de momentul în care se decretează că sistemul este defect. Astfel, durata de viață a unui sistem crossbar cu procesoare și memorii comune perfect fiabile și comutatoare expuse la defectare se definește ca intervalul de timp scurs de la punerea în funcțiune a sistemului până în momentul când:

- (I) nu mai există nici o CM la care să aibă acces toate PE
- (II) există două procesoare care nu mai pot comunica între ele
- (III) unul din procesoare se izolează complet de restul sistemului

Din cele trei definiții, prima este cea mai restrictivă, așa cum se poate observa pe figura 4.2.3.a: nici un PE nu este izolat, fiecare poate comunica cu oricare altul, dar toate nu mai au acces la aceeași memorie. Definiția (II) este mai restrictivă decât (III) (figura 4.2.3.b): încă PE<sub>1</sub> nu este izolat, dar nu poate comunica cu PE<sub>2</sub>. Evident un PE izolat duce conform fiecărei definiții la nefuncționarea sistemului, situația din figura 4.2.3.c prezentând cazul lui PE<sub>1</sub> izolat de restul sistemului.

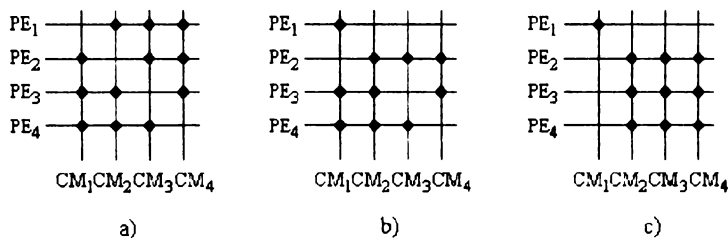


Figura 4.2.3.

Menționez că se pot concepe și alte definiții pentru durata de viață a sistemului crossbar. Astfel, în cazul unui sistem tolerant la defecte se poate admite că sistemul este funcțional până la izolarea unei anumite submulțimi a elementelor de procesare, adică, cu alte cuvinte, sistemul să fie funcțional până când un anumit număr minim de procesoare mai pot comunica (de tip  $k$ -din- $m$ ).

Pentru calculul timpului mediu de funcționare al sistemului ( $MTTF$ ) trebuie parcurse următoarele etape:

1. Se scrie funcția de structură  $\Phi$  a sistemului, conform definiției în cauză
2. Se calculează funcția de fiabilitate corespunzătoare  $R = \Pr\{\Phi=1\}$
3. Se înlocuiesc fiabilitățile componentelor (comutatoarelor) cu funcții de timp, obținându-se  $R(t)$
4. Se calculează valoarea așteptată  $E[t]$  a variabilei aleatoare având funcția de distribuție  $F(t) = 1 - R(t)$

În continuare voi ilustra cele de mai sus pe un exemplu, în scop comparativ. Consider un sistem crossbar cu  $n=m=3$ , pentru care îmi propun calcularea  $MTTF$  conform fiecărei variante a definiției (figura 4.2.4). Fiecare comutator are asociată câte o variabilă binară  $x_i$ , astfel încât  $x_i = 1$  dacă este funcțional și  $x_i = 0$  dacă este defect ( $i = 1, 2,$

..., 9). Probabilitățile de defectare ale comutatoarelor urmează o distribuție discretă de tip geometric, conform (4.2.4), fiabilitatea unui comutator fiind  $P = p^i$ .

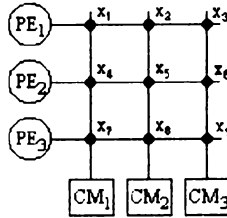


Figura 4.2.4.

Conform definiției (I), sistemul este funcțional dacă există cel puțin o coloană cu toate comutatoarele bune. În consecință, funcția booleană de structură va fi:

$$\Phi_1 = x_1x_4x_7 \cup x_2x_5x_8 \cup x_3x_6x_9 \tag{4.2.13}$$

Schema bloc de fiabilitate corespunzătoare acestei funcții este cea din figura 4.2.5. Pe baza ei se poate scrie imediat funcția de fiabilitate:

$$R_1 = 1 - (1 - P_B^3)^3 = 3P_B^3 - 3P_B^6 + P_B^9 \tag{4.2.14}$$

Având în vedere faptul că  $P_B = p^i$  și deci

$$R_1(t) = 3p^{3t} - 3p^{6t} + p^{9t} \tag{4.2.15}$$

valoarea așteptată a variabilei aleatoare, ceea ce este același lucru cu timpul mediu până la defectare, se poate exprima pe baza teoremei 3.3.2 ca fiind:

$$MTTF_1 = E[t] = 1 + 3 \frac{p^3}{1 - p^3} - 3 \frac{p^6}{1 - p^6} + \frac{p^9}{1 - p^9} \tag{4.2.16}$$

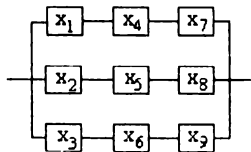


Figura 4.2.5.

Funcționarea corectă a sistemului conform definiției (II) se obține dacă, luând fiecare pereche de linii ale matricii de comutatoare, există cel puțin o coloană pe care ambele comutatoare sunt bune. Funcția booleană de structură va fi deci:

$$\Phi_2 = (x_1x_4 \cup x_2x_5 \cup x_3x_6) \cap (x_1x_7 \cup x_2x_8 \cup x_3x_9) \cap (x_4x_7 \cup x_5x_8 \cup x_6x_9) \tag{4.2.17}$$

Diagrama bloc de fiabilitate corespunzătoare lui  $\Phi_2$  este prezentată în figura 4.2.6. Rezolvarea ei nu mai este imediată, ci necesită aplicarea uneia din metodele prezentate la paragraful 3.5, deoarece același element se regăsește de mai multe ori pe schemă. Aici voi determina funcția de fiabilitate a sistemului pe baza funcției algebrice de structură, care pe baza figurii 4.2.6 este:

$$\begin{aligned} \Phi_2 = & [1 - (1 - x_1 x_4)(1 - x_2 x_5)(1 - x_3 x_6)][1 - (1 - x_1 x_7)(1 - x_2 x_8)(1 - x_3 x_9)][1 - (1 - x_4 x_7) \cdot \\ & \cdot (1 - x_5 x_8)(1 - x_6 x_9)] = x_1 x_4 x_7 + x_2 x_5 x_8 + x_3 x_6 x_9 + x_2 x_3 x_4 x_6 x_7 x_8 + x_2 x_3 x_4 x_5 x_7 x_9 - \\ & - x_1 x_2 x_3 x_4 x_6 x_7 x_9 - x_1 x_2 x_4 x_5 x_7 x_8 + x_1 x_2 x_5 x_6 x_7 x_8 + x_1 x_3 x_4 x_6 x_7 x_9 - x_2 x_3 x_5 x_6 x_8 x_9 + \\ & + x_1 x_2 x_5 x_6 x_7 x_9 + x_1 x_2 x_4 x_6 x_8 x_9 - x_1 x_2 x_3 x_4 x_6 x_7 x_8 - x_2 x_3 x_4 x_5 x_6 x_7 x_8 - x_1 x_2 x_3 x_4 x_5 x_7 x_9 - \\ & - x_2 x_3 x_4 x_5 x_6 x_7 x_9 - x_2 x_3 x_4 x_5 x_7 x_8 x_9 - x_2 x_3 x_4 x_6 x_7 x_8 x_9 - x_1 x_3 x_4 x_5 x_6 x_7 x_8 - x_1 x_2 x_3 x_5 x_6 x_7 x_8 - \\ & - x_1 x_2 x_3 x_4 x_5 x_8 x_9 - x_1 x_3 x_4 x_5 x_6 x_7 x_8 x_9 - x_1 x_2 x_3 x_4 x_5 x_7 x_8 x_9 - x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 + \\ & + 2x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 + 2x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_9 + 2x_1 x_2 x_3 x_4 x_5 x_7 x_8 x_9 + 2x_1 x_2 x_3 x_4 x_6 x_7 x_8 x_9 + \\ & + 2x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 + 2x_1 x_2 x_4 x_5 x_6 x_7 x_8 x_9 + 2x_1 x_2 x_3 x_4 x_5 x_6 x_8 x_9 + 2x_1 x_3 x_4 x_5 x_6 x_7 x_8 x_9 + \\ & + 2x_1 x_2 x_3 x_5 x_6 x_7 x_8 x_9 - 5x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 \end{aligned} \quad (4.2.18)$$

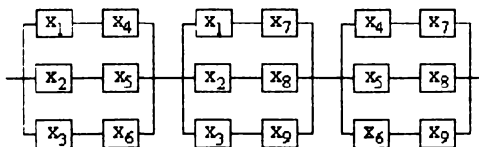


Figura 4.2.6.

Considerând comutatoare identice astfel încât  $\Pr\{x_i=1\} = \dots = \Pr\{x_9=1\} = P_B$  se obține expresia polinomială a fiabilității:

$$R_2 = 3P_B^3 + 3P_B^6 - 18P_B^7 + 18P_B^8 - 5P_B^9 \quad (4.2.19)$$

Pentru a afla valoarea medie a variabilei aleatoare, având în vedere distribuția geometrică a fiabilității componentelor, se utilizează teorema 3.3.2 :

$$MTTF_2 = E[t] = 1 + 3 \frac{p^3}{1-p^3} + 3 \frac{p^6}{1-p^6} - 18 \frac{p^7}{1-p^7} + 18 \frac{p^8}{1-p^8} - 5 \frac{p^9}{1-p^9} \quad (4.2.20)$$

În cazul calculului  $MTTF$  pe baza definiției (III), sistemul va fi considerat ca funcțional până când pe fiecare linie a matricei de comutatoare există cel puțin un comutator bun astfel încât pe aceeași coloană să mai existe cel puțin un comutator bun pe o altă linie. Funcția de structură corespunzătoare, exprimată sub formă booleană, va fi:

$$\begin{aligned} \Phi_2 = & [x_1(x_4 \cup x_7) \cup x_2(x_5 \cup x_8) \cup x_3(x_6 \cup x_9)] \cap [x_4(x_1 \cup x_7) \cup x_5(x_2 \cup x_8) \cup \\ & \cup x_6(x_3 \cup x_9)] \cap [x_7(x_1 \cup x_4) \cup x_8(x_2 \cup x_5) \cup x_9(x_3 \cup x_6)] = \\ & = [(x_1 x_4 \cup x_2 x_5 \cup x_3 x_6) \cap (x_1 x_7 \cup x_2 x_8 \cup x_3 x_9)] \cup [(x_1 x_4 \cup x_2 x_5 \cup x_3 x_6) \cap \\ & \cap (x_4 x_7 \cup x_5 x_8 \cup x_6 x_9)] \cap [(x_1 x_7 \cup x_2 x_8 \cup x_3 x_9) \cap (x_4 x_7 \cup x_5 x_8 \cup x_6 x_9)] \end{aligned} \quad (4.2.21)$$

Schema bloc de fiabilitate aferentă este dată în figura 4.2.7, pe baza căreia se poate scrie funcția booleană de structură ca fiind:

$$\Phi_3 = 1 - \{1 - (1 - x_1 x_4)(1 - x_2 x_5)(1 - x_3 x_6)\} [1 - (1 - x_1 x_7)(1 - x_2 x_8)(1 - x_3 x_9)] \cdot \\ \cdot \{1 - [1 - (1 - x_1 x_4)(1 - x_2 x_5)(1 - x_3 x_6)] [1 - (1 - x_4 x_7)(1 - x_5 x_8)(1 - x_6 x_9)]\} \cdot \\ \cdot \{1 - [1 - (1 - x_1 x_7)(1 - x_2 x_8)(1 - x_3 x_9)] [1 - (1 - x_4 x_7)(1 - x_5 x_8)(1 - x_6 x_9)]\} \quad (4.2.22)$$

Rezolvarea urmează aceeași procedură ca la  $\Phi_2$ , însă numărul de termeni fiind aici relativ mare, din motive de spațiu voi prezenta numai expresia polinomială de fiabilitate, care este:

$$R_3 = 3P_B^3 + 18P_B^4 - 36P_B^5 - 15P_B^6 + 72P_B^7 - 54P_B^8 + 13P_B^9 \quad (4.2.23)$$

Aplicând și aici teorema 3.3.2 pentru calculul timpului mediu până la defectare, se obține:

$$MTTF_3 = E[t] = 1 + 3 \frac{p^3}{1-p^3} + 18 \frac{p^4}{1-p^4} - 36 \frac{p^5}{1-p^5} - 15 \frac{p^6}{1-p^6} + \\ + 72 \frac{p^7}{1-p^7} - 54 \frac{p^8}{1-p^8} + 13 \frac{p^9}{1-p^9} \quad (4.2.24)$$

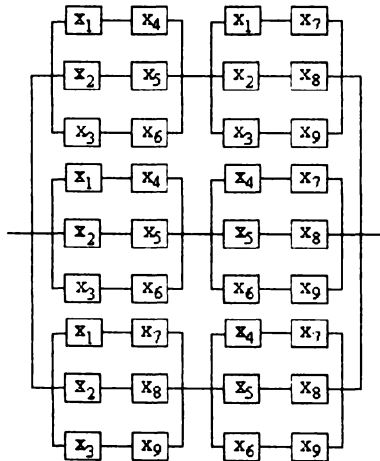


Figura 4.2.7.

Rezultatele date de relațiile (4.2.16), (4.2.20) și (4.2.24) pentru diferite valori ale lui  $p$  în cazul sistemului considerat sunt grupate în tabelul 4.2.8. Valorile numerice confirmă previziunile, și anume faptul că durata de viață a sistemului calculată pe baza definiției (I) este cea mai redusă. Celelalte două valori ale lui  $MTTF$  nu prezintă însă diferențe semnificative, însă cum s-a văzut efortul de calcul pentru determinarea lui  $MTTF_3$  este substanțial mai mare decât cel pentru  $MTTF_2$ . Mai este interesant faptul că o



reducere cu 0,9% a fiabilității comutatoarelor duce aproximativ la diminuarea de 10 ori a duratei de viață a sistemului.

$n=m=3$

$p$	$MTTF_1$	$MTTF_2$	$MTTF_3$
0,990	61,3	62,4	78,1
0,995	122,4	124,7	156,1
0,999	611,2	623,1	780,2

$MTTF$  in unități de timp

Tabelul 4.2.8.

În concluzie, consider că având în vedere și efortul de calcul, cel mai util indicator pentru calculul duratei de viață al sistemului crossbar este cel care consideră sistemul funcțional până când există cel puțin o memorie comună prin care să poată comunica toate procesoarele (definiția (I)). Acesta este și cel mai semnificativ din punct de vedere practic dacă ne gândim la funcționarea sistemului multiprocesor și are o formă analitică concisă de exprimare pornind de la funcția de fiabilitate [KAKAS96a]:

$$R = 1 - (1 - P_B^n)^m = 1 - \sum_{i=0}^m (-1)^i \binom{m}{i} P_B^{ni} = \sum_{i=1}^m (-1)^{i-1} \binom{m}{i} P_B^{ni} \quad (4.2.25)$$

$$MTTF_1 = E[t] = 1 + \sum_{i=1}^m (-1)^{i-1} \binom{m}{i} \frac{P^n}{1 - P^n} \quad (4.2.26)$$

O mărime care poate caracteriza performanțele unui sistem multiprocesor este timpul de răspuns  $T_{\bar{r}}$  (response time) la o cerere de acces lansată de un procesor la o memorie comună. Știind că  $P_A$  reprezintă probabilitatea ca o cerere de acces să fie acceptată, rezultă că  $1/P_A$  reprezintă numărul de cereri la memorie pentru ca cererea respectivă să fie acceptată (incluzând și cererea acceptată). Dacă se consideră numai cererile neservite dinaintea acceptării, atunci numărul mediu de unități de timp (sloturi) până când o cerere emisă de un PE către CM este acceptată, e dată de:

$$T_{\bar{r}} = \frac{1}{P_A} - 1 \quad (4.2.27)$$

În această relație,  $P_A$  se poate calcula pe baza (4.2.11), dar o soluție mai exactă se poate obține utilizând în aceasta rata dinamică a cererilor de acces. Dacă o cerere este acceptată imediat de memorie ( $P_A=1$ ), atunci  $T_{\bar{r}}=0$ .

Pe baza modelului prezentat pentru studiul caracteristicilor reunite de performanță și fiabilitate pentru un sistem crossbar se pot deduce relațiile care permit analiza comportării unui sistem multiprocesor cu magistrală unică (figura 2.2.1). Și aici este adecvată considerarea unui spațiu discret al timpului, presupunerile de la sistemul crossbar fiind valabile, cu observația că numărul de module CM se reduce la unul singur ( $m=1$ ). În lipsa comutatoarelor, elementul expus la defectare este magistrala sistem, conform funcției de fiabilitate dată de (4.2.4). De asemenea sunt valabile relațiile (4.2.1), (4.2.2) și (4.2.5), trebuie doar evaluată probabilitatea  $P_r = \Pr\{\text{să existe încă } k \text{ cereri la modulul CM}\}$ .

Aceasta se poate calcula ca și probabilitatea de a avea  $k$  încercări reușite din  $n-1$  corespunzătoare restului de procesoare, fiecare cu probabilitatea  $\alpha$ , adică, conform unei distribuții binomiale:

$$P_k = \binom{n-1}{k} \alpha^k (1-\alpha)^{n-1-k} \quad (4.2.28)$$

Ținând cont de acest rezultat, probabilitatea ca cererea emisă de un PE să fie reușită este, pe baza (4.1.2):

$$P_A = p^t \sum_{k=0}^{n-1} \frac{1}{k+1} \binom{n-1}{k} \alpha^k (1-\alpha)^{n-1-k} \quad (4.2.29)$$

Probabilitatea ca să existe o cerere de acces reușită la CM din partea unui PE va fi dată de  $\alpha P_A$ , adică:

$$P = \alpha p^t \sum_{k=0}^{n-1} \frac{1}{k+1} \binom{n-1}{k} \alpha^k (1-\alpha)^{n-1-k} \quad (4.2.30)$$

Această relație se poate aduce la o formă concisă dacă se ține cont de dezvoltarea binomială:

$$\begin{aligned} [\alpha + (1-\alpha)]^n &= \sum_{i=0}^n \binom{n}{i} \alpha^{n-i} (1-\alpha)^i = \sum_{i=0}^{n-1} \binom{n}{i} \alpha^{n-i} (1-\alpha)^i + (1-\alpha)^n = \\ &= \sum_{i=0}^{n-1} \frac{n}{n-i} \binom{n-1}{i} \alpha^{n-i} (1-\alpha)^i + (1-\alpha)^n = \\ &= n\alpha \sum_{i=0}^{n-1} \frac{1}{n-i} \binom{n-1}{i} \alpha^{n-i-1} (1-\alpha)^i + (1-\alpha)^n \end{aligned} \quad (4.2.31)$$

Efectuând schimbarea de variabilă  $k = n-1-i$  se obține:

$$1 - (1-\alpha)^n = n\alpha \sum_{k=0}^{n-1} \frac{1}{1+k} \binom{n-1}{k} \alpha^k (1-\alpha)^{n-1-k} \quad (4.2.32)$$

Exprimând suma din (4.2.32) și înlocuind-o în (4.2.30) rezultă probabilitatea unei cereri de acces reușite, scrisă sub o altă formă:

$$P = \alpha p^t \frac{1 - (1-\alpha)^n}{n\alpha} = p^t \frac{1 - (1-\alpha)^n}{n} \quad (4.2.33)$$

Ca o verificare a corectitudinii, pentru magistrala perfect fiabilă ( $p=1$ ), lățimea de bandă a sistemului care este dată de numărul așteptat de accese reușite

$$\beta = nP = p^t [1 - (1-\alpha)^n] \quad (4.2.34)$$

va avea tocmai expresia (4.1.20). Evident că și la acest tip de sistem, ca de altfel la toate considerate în acest capitol, dacă în relațiile anterioare se folosește o rată dinamică a cererilor în loc de cea statică se obține un model mai apropiat de realitate.

În cazul sistemelor multiprocesor cu magistrale multiple (figura 2.2.2), numărul de procesoare care obțin accesul la memoria comună este limitat atât de coliziunile cererilor de la nivelul modulelor CM, cât și de numărul căilor de comunicație, adică al magistralelor. În plus, dacă acestea din urmă sunt expuse la defectare, trebuie luat în calcul numai numărul de magistrale disponibile.

În prima etapă nu iau în considerare ca factor limitator decât interferențele cauzate de cererile simultane la modulele CM. Probabilitatea ca cererea emisă de un PE la un CM să fie reușită este dată, similar cu (4.2.5) de:

$$P_C = \sum_{k=0}^{n-1} \frac{1}{k+1} P_k \quad (4.2.35)$$

unde  $P_k = \Pr\{\text{există încă } k \text{ cereri la modulul CM}\}$ . Aceasta se poate evalua pe baza raționamentului expus la rețeaua crossbar, admitând că probabilitatea ca un anumit modul CM să fie accesat de un anumit PE este  $\alpha/m$ . Ca urmare, conform regulilor unei distribuții binomiale a numărului de procesoare rămase care cer acces la modulul CM, se obține  $P_k$  ca:

$$P_k = \binom{n-1}{k} \left(\frac{\alpha}{m}\right)^k \left(1 - \frac{\alpha}{m}\right)^{n-1-k} \quad (4.2.36)$$

În continuare trebuie calculat  $g(i) = \Pr\{\text{există exact } i \text{ cereri de acces reușite}\}$ , ceea ce se poate face în mod distinct în funcție de raportul dintre  $n$  și  $m$ . Acest lucru este așa, deoarece pentru  $n > m$  valoarea  $g(i)$  nu are semnificație fizică pentru  $i = m+1, \dots, n$ , numărul de cereri servite fiind limitat de numărul de module CM. La fel, pentru  $n < m$  nu are sens de exemplu valoarea  $g(m)$  dacă în total sunt numai  $n$  procesoare. În consecință, trebuie tratate separat cazurile  $n \leq m$  și  $n \geq m$ .

a)  $n \leq m$

Probabilitatea de a avea  $i$  reușite se poate calcula ca  $g(i) = \Pr\{\text{numărul de PE cu accesul reușit este } i\}$ , știind că probabilitatea ca un procesor să emită o cerere și aceasta să fie reușită este:

$$P = \alpha P_C \quad (4.2.37)$$

Probabilitatea ca să existe exact  $i$  reușite din  $n$  procesoare, fiecare cu probabilitatea  $P$ , este dată de funcția de masă de probabilitate a distribuției binomiale aferente, și anume:

$$g(i) = \binom{n}{i} P^i (1-P)^{n-i} \quad (4.2.38)$$

Fiind în total  $n$  procesoare, notez  $g'(i) = \Pr\{\text{există cel puțin } i \text{ cereri de acces reușite}\}$ , valoarea acesteia fiind:

$$g'(i) = g(i) + g(i+1) + \dots + g(n) \quad (4.2.39)$$

În a doua etapă țin cont de faptul că pentru ca cele  $i$  cereri selectate în prima etapă să fie într-adevăr reușite, este necesar ca să existe cel puțin  $i$  magistrale disponibile (funcționale), dar numărul de cereri servite nu poate depăși numărul magistrarelor din sistem ( $b$ ). De asemenea vor exista  $i$  reușite și în cazul în care există mai mult de  $i$  cereri selectate, dar numărul magistrarelor bune este  $i$ . Dacă notez  $e(i) = \Pr\{\text{există exact } i \text{ magistrale bune}\}$  și  $e'(i) = \Pr\{\text{cel puțin } i \text{ magistrale sunt bune}\}$ , atunci numărul mediu de procesoare care vor avea cererile la memoria comună servite se poate exprima sub forma:

$$\begin{aligned} \beta = & 1[g(1)e'(1) + g'(2)e(1)] + 2[g(2)e'(2) + g'(3)e(2)] + \\ & + \dots + b[g(b)e'(b) + g'(b+1)e(b)] \end{aligned} \quad (4.2.40)$$

Trebuie determinate probabilitățile  $e(i)$  și  $e'(i)$ , presupunând că probabilitatea ca o magistrală să fie funcțională este egală, conform (4.2.4), cu  $P_B = p'$  (unde  $p=1-q$ ,  $q$  fiind rata de defectare pe slot). Probabilitatea ca să existe exact  $i$  magistrale bune este în consecință dată de:

$$e'(i) = \binom{b}{i} P_B^i (1 - P_B)^{b-i} \quad (4.2.41)$$

Dacă  $x$  reprezintă variabila aleatoare corespunzătoare numărului de magistrale bune, atunci probabilitatea ca cel puțin  $i$  magistrale din  $b$  să fie funcționale se va putea exprima ca și:

$$e'(i) = \Pr\{x \geq i\} = 1 - \Pr\{x < i\} = 1 - \sum_{j=0}^{i-1} e(j) = 1 - \sum_{j=0}^{i-1} \binom{b}{j} P_B^j (1 - P_B)^{b-j} \quad (4.2.42)$$

În final, se poate determina probabilitatea  $P_A$  ca o cerere emisă de un procesor la memoria comună să fie servită, având în vedere atât limitările structurale, cât și cele de fiabilitate. Pe baza relației (4.1.24), în care  $\beta$  se înlocuiește cu expresia (4.2.40), probabilitatea unui acces reușit va fi:

$$P_A = \sum_{i=1}^b \left[ \frac{i}{n\alpha} g(i)e'(i) + \frac{i}{n\alpha} g'(i+1)e(i) \right] = \sum_{i=1}^b [h(i)e'(i) + h'(i+1)e(i)] \quad (4.2.43)$$

Am pus relația (4.2.43) sub această formă, pentru a pune în evidență faptul că  $h(i)$  și  $h'(i)$  exprimă probabilitatea ca exact  $i$  procesoare, respectiv  $i+1$  procesoare, să aibă cererile selectate pentru a fi servite în prima etapă, printre ele fiind și cererea procesorului în cauză (pentru care se exprimă  $P_A$ ).

(Pentru magistrale perfect fiabile este evident că relația (4.2.40) este similară cu (4.1.23), dar însumarea se face până la  $n$  în loc de  $m$ ).

b)  $n \geq m$

În acest caz este mai convenabil a evalua probabilitatea de a avea  $i$  reușite sub forma  $g(i) = \Pr\{\text{există exact } i \text{ module CM diferite care au cereri}\}$ . Ca și la rețeaua crossbar, probabilitatea ca un modul CM să aibă cel puțin o cerere este:

$$r = 1 - \left(1 - \frac{\alpha}{m}\right)^n \quad (4.2.44)$$

Fiind în total  $m$  astfel de module, probabilitatea căutată este:

$$g(i) = \binom{m}{i} r^i (1-r)^{m-i} \quad (4.2.45)$$

Și aici notez cu  $g'(i) = \Pr\{\text{există cel puțin } i \text{ cereri de acces reușite}\}$ , dar întrucât numărul maxim de cereri reușite nu poate depăși numărul de module CM, valoarea ei este:

$$g'(i) = g(i) + g(i+1) + \dots + g(m) \quad (4.2.46)$$

Urmând același raționament ca la cazul a), în a doua etapă se poate exprima numărul mediu de memorii comune ocupate, ținând cont și de limitarea impusă de numărul de magistrale, precum și de fiabilitatea magistralelor. Relația de calcul este aceeași cu (4.2.40), iar probabilitatea ca cererea emisă de un procesor la memoria comună să fie reușită se obține pe baza relației (4.2.43). Mai este de subliniat faptul că relațiile de calcul pentru  $\beta$  și  $P_A$  sunt identice, dar pentru două situații structurale diferite, pentru care  $g(i)$  și  $g'(i)$  se calculează în mod distinct, de aceea trebuie utilizate după caz. De asemenea, ca o verificare se poate ușor constata că pentru  $m=b=1$  relațiile precedente se reduc la cele deduse la sistemul cu magistrală unică.

În cazul sistemelor cu rețea multinivel (MIN), rezultatele obținute la tratarea rețelelor crossbar pot fi de asemenea folosite, având în vedere faptul că elementele de comutare al rețelei MIN sunt de fapt niște comutatoare crossbar cu  $a$  intrări și  $b$  ieșiri (figura 2.4.7). Voi admite într-un spațiu discret al timpului aceeași rată de defectare  $q=1-p$  pe un slot pentru oricare legătură de tip intrare-ieșire a unui element de comutare, acestea din urmă fiind presupuse identice (rețea omogenă). Întrucât sistemul se compune din  $N$  nivele, trebuie mai întâi determinată rata cererilor ( $r_j$ ) pe o ieșire a elementului de comutare aflat într-un nivel intermediar ( $j$ ), presupunând că pe intrările sale există o rată de cereri  $r_{j-1}$ . Aceasta se poate calcula cunoscând probabilitatea ca o cerere aplicată pe o intrare a comutatorului să fie reușită (adică transmisă pe ieșire, ținând cont de posibilitatea defectării legăturilor interne și de conflictele cu alte cereri simultane), care este dată de relația (4.2.11) și în care să înlocuiesc  $n$  cu  $a$ ,  $m$  cu  $b$  și  $\alpha$  cu  $r_{j-1}$ . Urmează că rata pe o ieșire este egală cu numărul de cereri acceptate (și transmise prin comutator), raportat la numărul total de ieșiri, adică:

$$r_j = \frac{ar_{j-1}}{b} p^i \sum_{k=0}^{a-1} \sum_{i=0}^{a-k-1} \frac{1}{k+1} \left( \frac{a-1}{k+i} \frac{r_{j-1}}{b} \right)^{k+i} \left( 1 - \frac{r_{j-1}}{b} \right)^{a-1-(k+i)} \binom{k+i}{k} p^{ik} (1-p)^i \quad (4.2.47)$$

Expresia obținută este o relație de recurență care trebuie aplicată pentru toate nivelele rețelei MIN ( $j = 1, 2, \dots, N$ ), admitând pe intare rata  $r_0 = \alpha$  (probabilitatea ca un procesor conectat pe o intrare să ermită o cerere de acces la memoria comună în slotul care urmează). Se obține în final  $r_N$ , adică probabilitatea ca o cerere să fie transmisă la o ieșire a ultimului nivel. Rezultă că dacă rețeaua are  $m$  ieșiri pe care sunt conectate modulele CM, atunci lățimea de bandă a sistemului va fi

$$\beta = m r_N \quad (4.2.48)$$

iar probabilitatea ca o cerere de acces la memoria comună din partea unui PE conectat pe una din cele  $n$  intrări să fie acceptată, este dată de:

$$P_A = \frac{m r_N}{n\alpha} \quad (4.2.49)$$

Dacă se consideră că rețeaua este lipsită de nivele suplimentare, având câte o singură cale de la fiecare intrare la fiecare ieșire, atunci  $n = a^N$  și  $m = b^N$ . În mod uzual, comutatoarele rețelei MIN sunt celule de tip  $2 \times 2$  (rețele baseline, Omega, etc.) și rețelele sunt rectangulare ( $n=m=2^N$ ). În acest caz se pot deduce relații particulare pentru indicatorii de performanță. Comutatorul, asemuit cu un crossbar cu două intrări și două ieșiri, conține patru elemente expuse la defectare, cu aceeași probabilitate (figura 4.2.9). Aceste elemente corespund la câte o legătură pe graful asociat, defectarea fiecăruia ducând la întreruperea comunicației dintre intrarea și ieșirea corespunzătoare. Într-un caz concret, identificarea lor ține de modul de construcție al comutatorului.

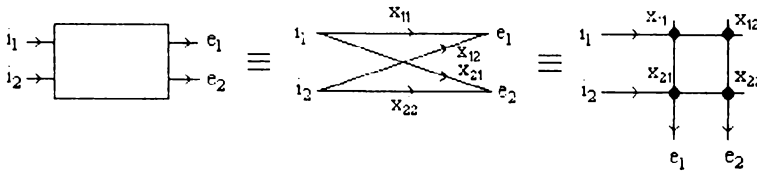


Figura 4.2.9.

Luând în (4.2.47)  $a=b=2$ , după efectuarea calculelor se obține relația de recurență simplă:

$$r_j = r_{j-1} p^j \left( 1 - \frac{r_{j-1} p^j}{4} \right) \quad (4.2.50)$$

Aceasta se poate pune ușor și sub forma

$$r_j = 1 - \left( 1 - \frac{r_{j-1}}{2} p^j \right)^2 \quad (4.2.51)$$

care este comparabilă cu (4.1.26) și o verifică pentru comutatoare perfecte ( $p=1$ ). Relația (4.2.51) permite următoarea interpretare: termenul  $r_{j-1} p^j / 2$  reprezintă probabilitatea ca o

ieșire a comutatorului să primească cerere de la una din intrări și calea corespunzătoare să fie funcțională. Rețeaua are aici  $N = \log_2 n$  nivele, deci  $r_N$  se obține prin aplicarea relației de recurență succesiv pentru  $j = 1, 2, \dots, \log_2 n$ . Lățimea de bandă, respectiv probabilitatea unui acces reușit prin rețea, vor fi date la aceste tipuri particulare de rețele de:

$$\beta = 2 r_N \quad (4.2.52)$$

$$P_A = \frac{1}{\alpha} r_N \quad (4.2.53)$$

Este de menționat faptul că relațiile deduse sunt valabile și în cazul rețelelor dinamice (netolerante) pe un singur nivel, de tipul celor cu amestecare perfectă (perfect shuffle). Dacă o astfel de rețea este în conexiune recirculantă, atunci evident aceste relații se vor considera succesiv la fiecare pas.

### 4.3. Rezultate experimentale

Unul din indicatorii de performanță cel mai des utilizat la sistemele multiprocesor este *lățimea de bandă* (bandwidth), adică numărul mediu de module de memorie CM ocupate ( $\beta$ ). Acest număr este egal și cu numărul mediu de procesoare care au făcut acces la memoria comună și accesul este reușit. Un procesor poate fi într-una din următoarele stări pe durata unui slot:

- activ, cu probabilitatea  $1 - \alpha$
- a cerut acces și acesta a fost reușit, cu probabilitatea  $\alpha \cdot P_A$
- a cerut acces și acesta a fost nereușit, cu probabilitatea  $\alpha (1 - P_A)$

Similar cu numărul de succese dintr-o secvență de încercări Bernoulli de probabilități egale, numărul de accese reușite pentru cele  $n$  procesoare reprezintă o variabilă aleatoare cu distribuție binomială. Probabilitatea de reușită dintr-o încercare fiind  $\alpha \cdot P_A$ , valoarea așteptată a variabilei aleatoare este:

$$\beta = n \cdot \alpha \cdot P_A \quad (4.3.1)$$

Pentru modelul considerat (cu posibilitatea de defectare a comutatoarelor),  $\beta$  este o funcție de numărul de unități de timp ( $t$ ). În figura 4.3.1 este reprezentată evoluția lui  $\beta$  până la 1000 de sloturi, pentru un sistem de tip crossbar cu  $n=15$ ,  $m=10$ ,  $\alpha=0,7$  și  $p=0,998$ . Pentru validarea modelului propus am elaborat în Turbo Pascal un program de simulare a comportării sistemului, simularea pe calculator fiind o metodă uzuală de investigare a performanțelor sistemelor paralele [TAYL94]. Acest program (descries în Anexa 4) actualizează la fiecare unitate de timp matricea de comutatoare ținând cont de rata de defectare  $q=1-p$ , determină care procesoare lansează cereri la memoria comună (cu rata  $\alpha$ ) și stabilește pe baza unei selecții uniforme modulul de memorie accesat de fiecare procesor. În conformitate cu un sistem fix de priorități între procesoare se determină care pot avea accesul reușit, condiționat însă de starea funcțională a comutatorului de pe calea de acces. În final se determină la fiecare slot numărul de procesoare cu accesul reușit și al celor în așteptare, iar apoi se efectuează media cumulată a acestor valori pentru toate iterațiile făcute. Rezultatele date de trei rulări ale acestui

program, cu increment de 50 de unități de timp, sunt reprezentate cu puncte pe figura 4.3.1. Se observă că rezultatele obținute prin simulare confirmă valabilitatea modelului matematic. (La un număr mare de iterații, peste 1000, se obține o suprapunere perfectă). O altă variantă a programului de simulare, mai apropiată de funcționarea reală, ține cont de reluarea de către un procesor a unei cereri neservite în următorul slot la același modul de memorie, în cazul în care refuzul s-a datorat unui conflict cu o altă cerere. Dacă neservirea s-a datorat comutatorului defect, atunci cererea nu mai este menținută în continuare. Media rezultatelor date de 100 de simulări este dată în figura 4.3.2 pentru un sistem cu  $n=10$ ,  $m=6$ ,  $\alpha=0,7$  și  $p=0,997$ , comparativ cu rezultatele calculate pe baza modelului matematic propus. Se poate remarca că și în acest caz curbele prezintă o bună suprapunere, ceea ce demonstrează justetea modelului pentru acest indice de performanță al sistemului.

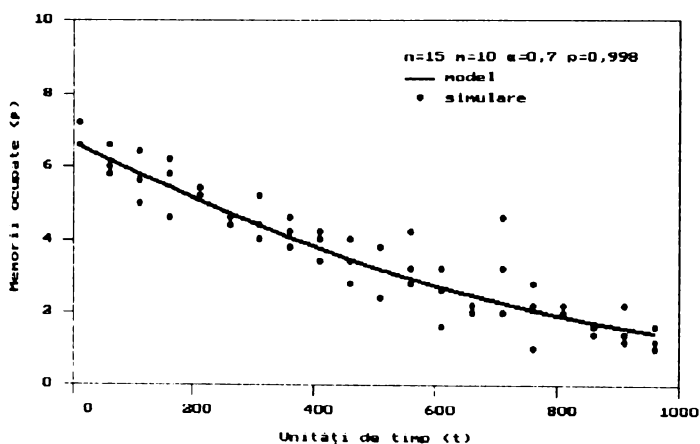


Figura 4.3.1.

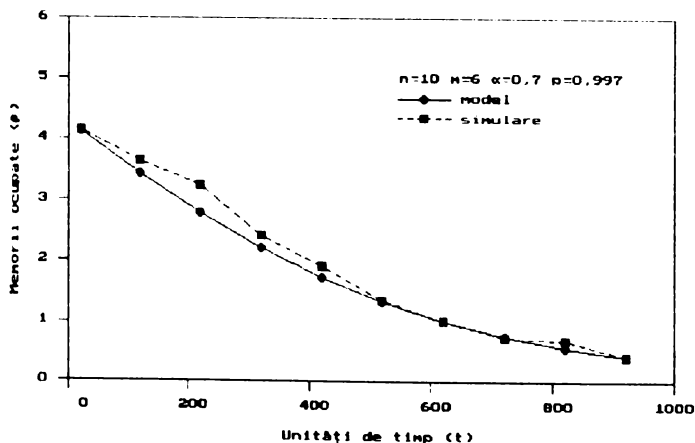


Figura 4.3.2.



Un indicator de contraperformață este numărul mediu de procesoare care stau ( $w$ ), adică au cerut acces dar acesta nu a fost reușit datorită comutatorului defect sau coliziunii cu o altă cerere:

$$w = n \cdot \alpha (1 - P_A) \quad (4.3.2)$$

Curbele comparative ale lui  $w$  pentru sistemul considerat mai înainte, obținute pe baza modelului și simulării, sunt date în figura 4.3.3. Și pentru acest indicator valorile calculate aproximează foarte bine pe cele experimentale rezultate prin simulare. În figura 4.3.4 este dată evoluția lui  $w$  în cazul unui sistem cu 15 procesoare și 10 memorii, pentru diferite valori ale lui  $p$ . Pentru  $p \neq 1$  curbele tind asimptotic spre valoarea  $n \cdot \alpha = 10,5$ , corespunzătoare cazului când toate comutatoarele s-au defectat. De pe curbe se poate citi aproximativ și viața utilă a sistemului.

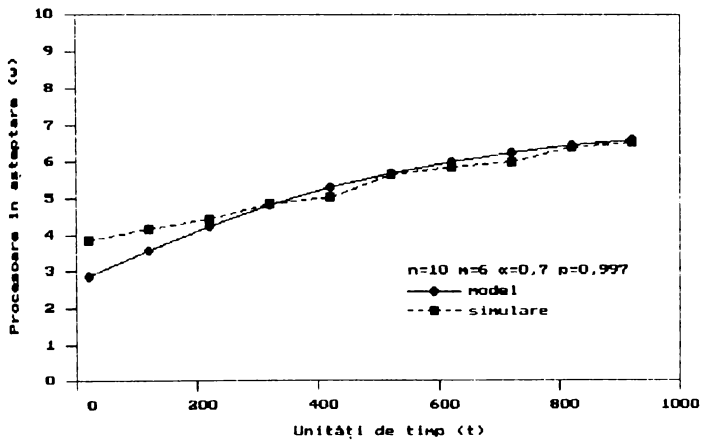


Figura 4.3.3.

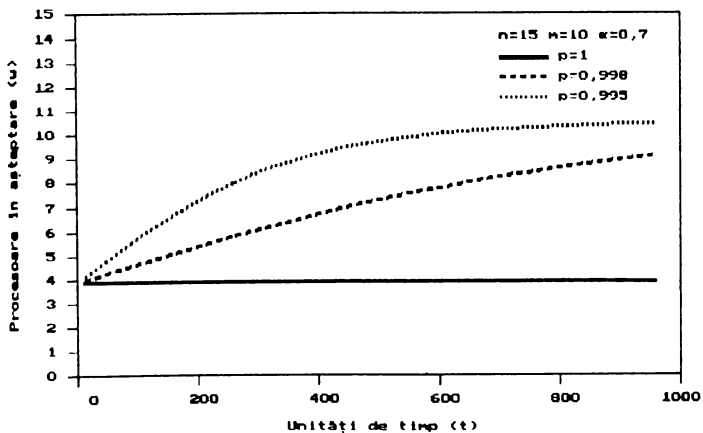


Figura 4.3.4.

În figura 4.3.5 am reprezentat variația lui  $w$  în funcție de probabilitatea unei cereri de acces la CM ( $\alpha$ ) pentru sistemul considerat, la diferite valori ale lui  $p$ . Se remarcă existența practică a unei probabilități de prag ( $p = 0,989$ ), sub care curbele devin liniare și se suprapun. Explicația este că numărul de procesoare în așteptare nu poate depăși valoarea  $n \cdot \alpha$ , iar cu probabilități de bună funcționare mai mici decât cea de prag, toate căile se blochează până la momentul  $t = 500$  considerat. Existența probabilității de prag se pune în evidență și mai bine pe graficul din figura 4.3.6, unde pe axa absciselor s-a reprezentat fiabilitatea comutatorului ( $p$ ).

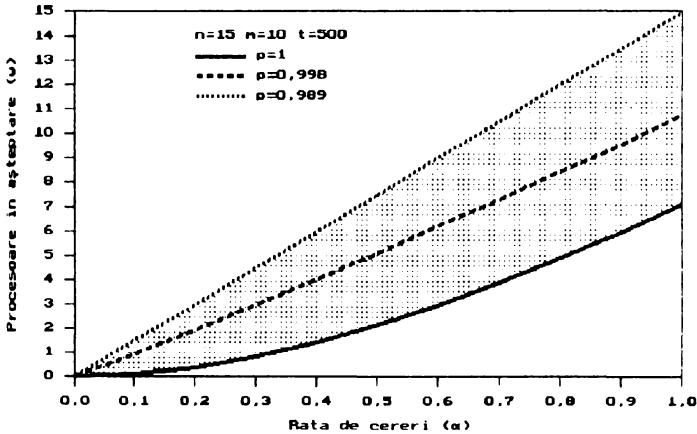


Figura 4.3.5.

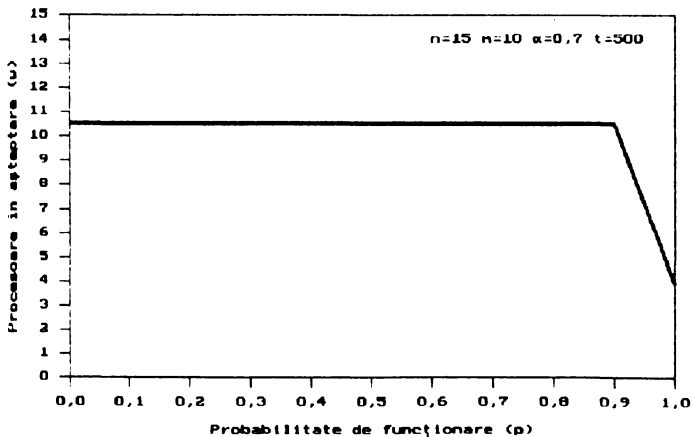


Figura 4.3.6.

Se pune problema de a găsi un indicator global de eficiență al sistemului, care să fie în ajutorul proiectantului de sistem. Consider că  $n$  este dat, prin natura aplicației, iar  $\alpha$  trebuie să fie cât mai mic (cât mai puține necesități de transfer de date între procesoare, autonomie cât mai mare). Întrebarea este de câte module CM este nevoie? Dacă  $m$  este mic, atunci prea multe PE vor fi în așteptare, iar dacă  $m$  este mare, atunci prea multe CM vor sta nefiind solicitate și crește inutil costul și dimensiunea sistemului. Deci trebuie să existe o valoare optimă. Pentru a o găsi, definesc *indicatorul de eficiență al sistemului* ( $\varepsilon$ ) ca fiind:

$$\varepsilon = \frac{\text{nr. PE care lucrează} \times \text{nr. CM ocupate}}{n \times m} = \frac{[n(1 - \alpha) + \beta]\beta}{nm} \quad (4.3.3)$$

Eficiența sistemului este reprezentată în figura 4.3.7 în funcție de  $m$ , pentru cazul comutatoarelor perfecte (timpul nu mai are astfel importanță), pentru diferite valori ale lui  $\alpha$ . Se remarcă faptul că pentru numărul dat de procesoare (aici 15) curbele prezintă câte un maxim, pentru care se poate citi numărul optim de module CM, ceea ce reprezintă o informație utilă proiectantului de sistem. Numărul de memorii necesare este mai mic pentru  $\alpha$  mic. Creșterea nejustificată a lui  $m$  nu duce la mărirea eficienței sistemului și peste o anumită valoare nu mai are practic influență.

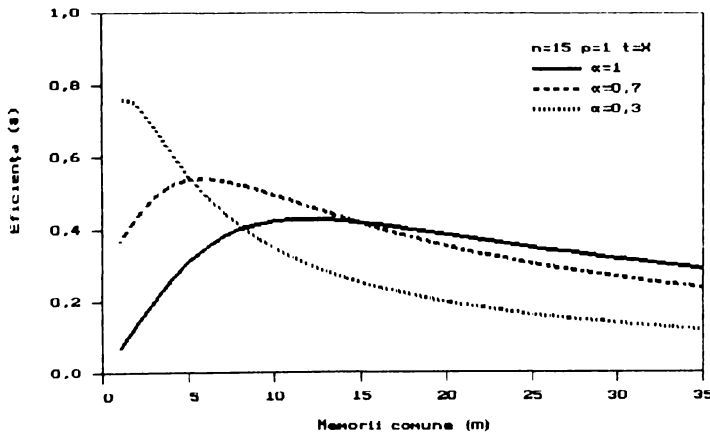


Figura 4.3.7.

În cazul unui sistem multiprocesor cu magistrală unică, care are  $n$  procesoare și magistrala este expusă la defectare, lățimea de bandă ( $\beta$ ) este dată de relația (4.2.34). Întrucât aici există doar o singură memorie comună accesibilă procesoarelor prin magistrala sistem, mărirea  $\beta$  exprimă mai degrabă gradul de ocupare al acestei memorii decât numărul mediu de module de memorie care lucrează. Valoarea lui  $\beta$  depinde în principal de numărul de procesoare din sistem și rata cererilor ( $\alpha$ ), în sensul că un număr mare de procesoare cu o rată de cereri ridicată duce la ocuparea aproape în permanență a

memoriei comune ( $\beta \approx 1$ ). Dacă intervine însă și probabilitatea de cădere a magistralei, ocuparea memoriei prezintă o scădere exponențială cu timpul pe baza relației menționate, lucru ilustrat în graficul din figura 4.3.8 pentru cazul  $n=5$ ,  $\alpha=0,7$  și  $p=0,998$ . Pe același grafic am reprezentat pentru comparație rezultatele date de execuția programului întocmit pentru simularea funcționării sistemului (prezentat în Anexa 4), numărul de iterații efectuate fiind de 100. Se poate constata că cele două curbe urmează aceeași tendință, ceea ce confirmă justetea modelului matematic propus.

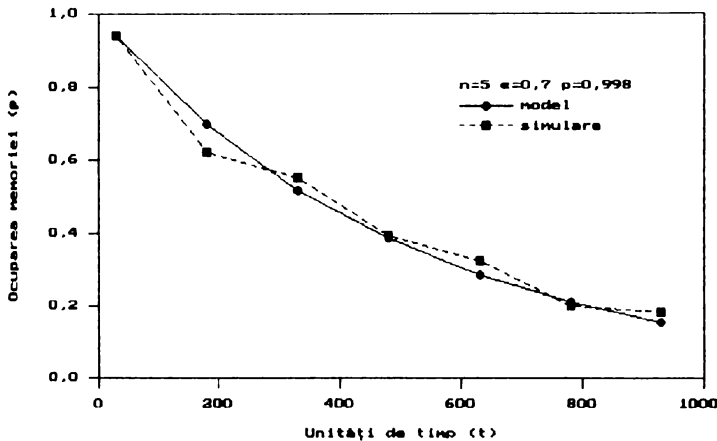


Figura 4.3.8.

Pentru studiul comportării sistemului considerat nu este lipsit de interes determinarea probabilității ca un anumit procesor să obțină un acces reușit ( $P_A$ ) dacă a lansat o cerere către memoria comună. Cunoscând pe  $\beta$ , egal cu numărul mediu de procesoare care primesc acces,  $P_A$  se poate calcula analitic raportând această valoare la numărul mediu de procesoare care au cerut acces ( $\alpha n$ ), adică:

$$P_A = \frac{\beta}{\alpha n} = \frac{p^t [1 - (1 - \alpha)^n]}{\alpha n} \quad (4.3.4)$$

Pe figura 4.3.9 am reprezentat graficul funcției (4.3.4) pentru cazul sistemului de mai înainte, luând ca variabilă numărul de unități de timp. Alături, pentru comparare, se află rezultatele obținute prin simulare. Se observă că evoluția curbelor este aceeași, dar în cazul acestui indicator ele sunt totuși mai depărtate, mai ales în prima porțiune, când probabilitatea ca magistrală să fie funcțională este mare. Explicația acestei diferențe (totuși acceptabile) constă în faptul că în programul de simulare a funcționării reale am ținut cont de faptul că o cerere neservită din cauza unei coliziuni cu o altă cerere simultană se va menține în continuare și la următoarea unitate de timp. Acest lucru duce la creșterea numărului de procesoare care cer acces și reduce probabilitatea de reușită față de situația modelată matematic.

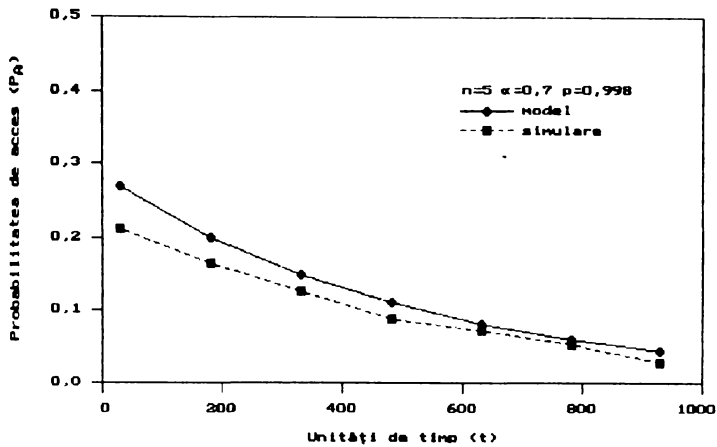


Figura 4.3.9.

Timpul de viață al sistemului va depinde numai de momentul în care se defectează magistrala sistemului. Dacă  $q=1-p$  reprezintă probabilitatea de defectare pe o unitate de timp (slot), atunci conform mediei unei distribuții geometrice numărul de sloturi până la defectare (inclusiv acesta) va fi:

$$MTTF = \frac{1}{q} = \frac{1}{1-p} \quad (4.3.5)$$

Graficul acestei funcții pentru  $p \in [0,990 ; 1]$  este reprezentat în figura 4.3.10. Pe baza acestei figuri se poate face remarcă importantă pentru proiectantul de sistem că timpul mediu până la defectare ( $MTTF$ ) crește vertiginos dacă rata de fiabilitate a liniilor de magistrală este superioară unei valori de aproximativ  $p > 0,998$ .

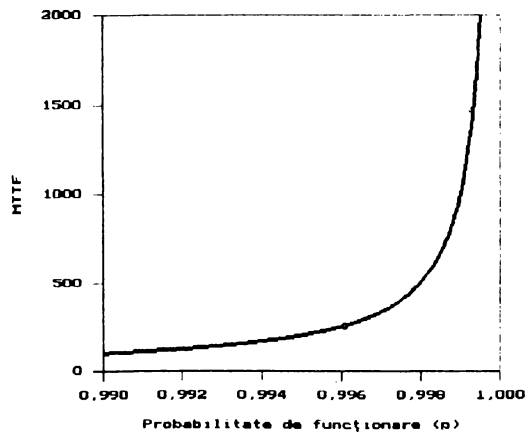


Figura 4.3.10.

O problemă deosebită la orice sistem multiprocesor cu magistrală unică o reprezintă evitarea saturării magistralei, deoarece dacă numărul de procesoare conectate în sistem ( $n$ ) depășește limita de saturație, atunci traficul mare de pe magistrală va duce la creșterea numărului de procesoare aflate în așteptare de a primi acces la magistrală. Dacă numărul de procesoare care stau este mare, atunci se pierde din eficiența dobândită prin distribuirea sarcinilor la mai multe procesoare. Pentru un sistem dat, considerând cazul ideal în care timpul de magistrală se împarte egal și succesiv între procesoare, numărul de procesoare aflate în așteptare ( $w$ ) depinde de gradul de ocupare al magistralei, care este de fapt  $\alpha$  (probabilitatea ca un procesor să emită cerere la memoria comună):

$$w = \begin{cases} 0 & \text{dacă } n \leq \frac{1}{\alpha} \\ n - \frac{1}{\alpha} & \text{dacă } n > \frac{1}{\alpha} \end{cases} \quad (4.3.6)$$

În figura 4.3.11. am reprezentat grafic pe  $w$  în funcție de numărul de procesoare din sistem, pentru diferite valori ale lui  $\alpha$  (fiabilitatea magistralei fiind considerată unitară), pe baza relației de la modelul propus la paragraful 4.2:

$$w = \alpha n (1 - P_A) \quad (4.3.7)$$

Se observă că aceste curbe sunt tangente la cele date de (4.3.6) pentru cazul ideal, ultimele fiind linii frânte în punctele care reprezintă pragurile de saturație  $n = n_s = 1/\alpha$  (valorile de prag fiind 2, 5, respectiv 10 pentru cele trei situații de pe figură). În realitate saturația apare deja la un număr de procesoare mai mic decât cel de prag, deoarece ele au o funcționare asincronă și nu se poate garanta ca unul să ceară magistrala exact în momentul când cel precedent renunță la ea. Curbele date de modelul propus reflectă această situație, fiind deci conforme unei funcționări reale a sistemului.

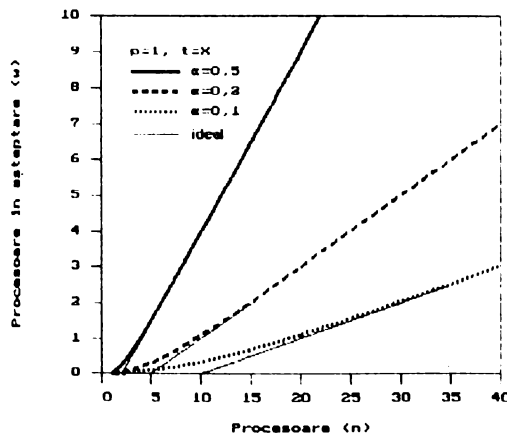


Figura 4.3.11.

La un sistem multiprocesor cu magistrale multiple, așa cum am arătat la 4.2, calculul numărului mediu de module de memorie comună ocupate ( $\beta$ ) diferă după cum numărul de procesoare ( $n$ ) este mai mare sau mai mic decât numărul acestor module ( $m$ ). De aceea, pentru validarea prin simulare a modelului propus (pe baza algoritmului din Anexa 4), am tratat distinct cele două situații. Graficele comparative, aproape suprapuse, ale rezultatelor obținute pe baza modelului (relația (4.2.40)) și prin simulare pentru un sistem cu  $n > m$  sunt prezentate în figura 4.3.12. Se poate constata că în momentul inițial, când toate magistralele sunt bune, curbele pornesc aproximativ de la valoarea  $\beta=3$  egală cu numărul magistralelor ( $b=3$ ), deoarece pentru rata mare admisă a cererilor ( $\alpha=0,9$ ) numărul celor care pot fi servite este limitat de numărul magistralelor de comunicație. Numărul de procesoare servite urmează apoi o scădere exponențială pe măsura defectării magistralelor.

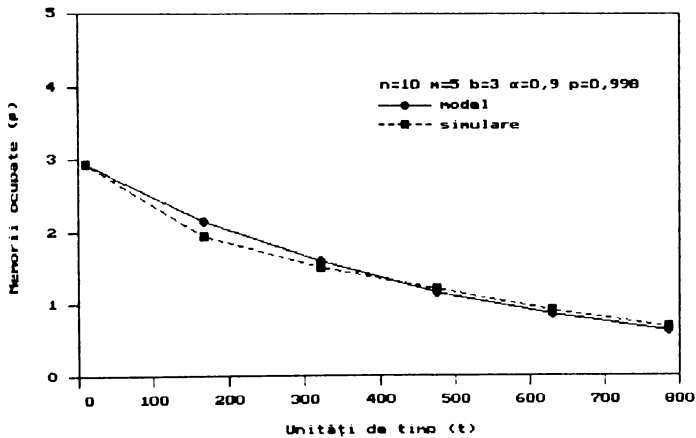


Figura 4.3.12.

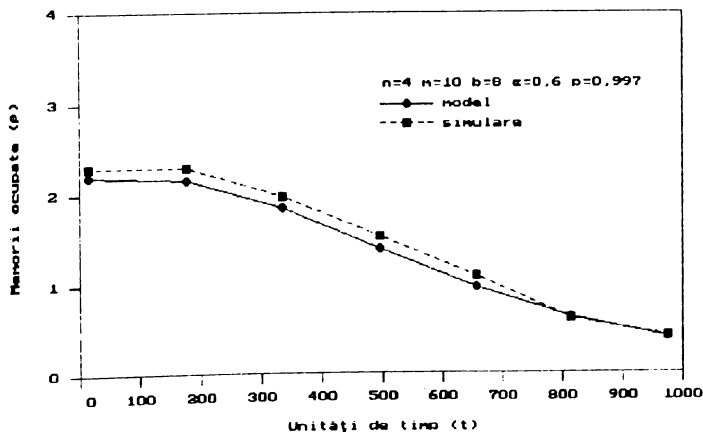


Figura 4.3.13.

Pentru cazul  $n < m$  am luat un număr mic de procesoare ( $n=4$ ) cu o rată mai mică a cererilor ( $\alpha=0,6$ ), tocmai pentru a ilustra faptul că limitarea numărului de cereri servite începe mai târziu, după ce o parte a magistralelor s-au defectat (figura 4.3.13). Numărul mediu al modulelor CM care au cereri se poate calcula cu relația (4.1.21) și este în acest caz de 2,19, ceea ce necesită doar 3 magistrale funcționale. De aceea curbele din figura 4.3.13, foarte apropiate pentru model și simulare, au până la aproximativ  $t=200$  o porțiune constantă, corespunzătoare situației când numărul mediu de cereri este mai mic decât numărul de magistrale disponibile. Se poate pune atunci următoarea întrebare: fiind dat un anumit sistem multiprocesor, care este numărul de magistrale necesare, pentru a nu crește exagerat și inutil costurile și complexitatea sistemului ?

Pentru a răspunde la întrebarea de mai înainte, trebuie determinat numărul mediu de cereri diferite emise de cele  $n$  procesoare la cele  $m$  module de memorie. Această valoare a fost deja dedusă, fiind dată de relația (4.1.21). Numărul de magistrale necesare pentru a servi toate aceste cereri se obține prin urmare pe baza expresiei:

$$b_n = \left\lceil m \left[ 1 - \left( 1 - \frac{\alpha}{m} \right)^n \right] \right\rceil \quad (4.3.8)$$

unde  $\lceil \cdot \rceil$  reprezintă rotunjirea în sus la următorul întreg. În figura 4.3.14 am reprezentat grafic această funcție pentru o configurație dată în funcție de rata cererilor la memoria comună, considerând magistrale perfect fiabile ( $p=1$ ). S-a obținut o funcție în trepte, de pe care proiectantul de sistem poate citi numărul optim de magistrale necesare în sistem, astfel încât să nu fie prevăzute prea multe magistrale ineficient exploatate raportat la numărul de cereri.

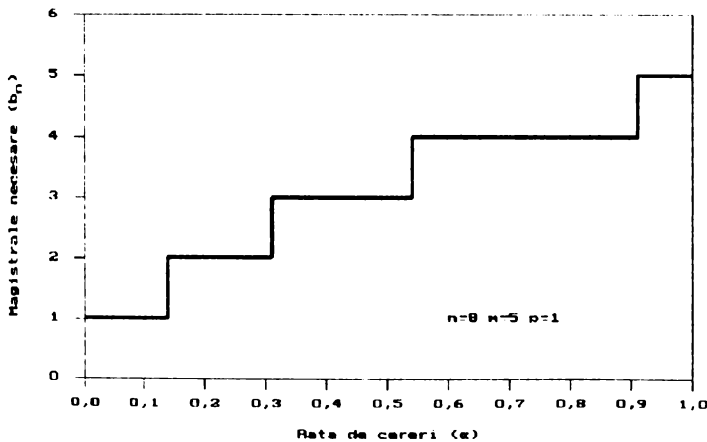


Figura 4.3.14.

Durata de viață a unui sistem cu magistrale multiple ține până în momentul în care se defectează toate magistralele. Din punct de vedere al schemei de fiabilitate, magistralele sistemului pot fi considerate conectate în paralel, adică sistemul este funcțional dacă există



cel puțin o magistrală bună. Fiecare magistrală având conform (4.2.4) fiabilitatea  $P_B = p^b$ , fiabilitatea sistemului cu  $b$  magistrale este dată de:

$$R = 1 - (1 - P_B)^b = \sum_{i=1}^b (-1)^{i-1} \binom{b}{i} p^{it} \quad (4.3.9)$$

Timpul mediu până la defectare ( $MTTF$ ) corespunzător acestei funcții polinomiale se obține pe baza teoremei 3.3.2, și anume:

$$MTTF = 1 + \sum_{i=1}^b (-1)^{i-1} \binom{b}{i} \frac{p^i}{1 - p^i} \quad (4.3.10)$$

Pe figura 4.3.15 am reprezentat grafic  $MTTF$  în funcție de rata de fiabilitate a unei magistrale ( $p$ ) având ca parametru numărul de magistrale din sistem. Se observă, ca și la sistemele cu magistrală unică o creștere substanțială a duratei de viață pentru  $p$  apropiat de 1. O dublare a numărului de magistrale nu duce însă decât la o creștere neînsemnată a timpului de viață.

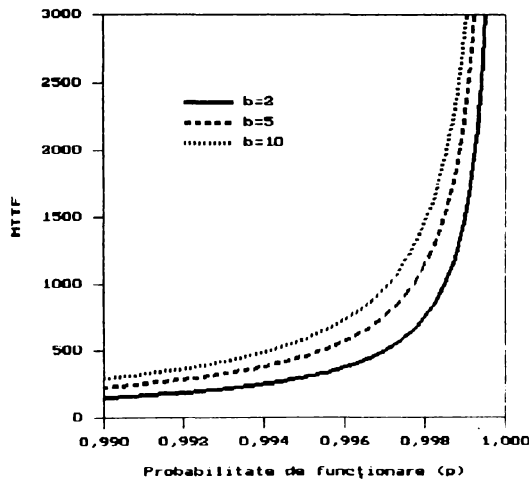


Figura 4.3.15.

Pentru studiul comportării unui sistem multiprocesor cu rețea multinivel am ales o rețea rectangulară de tip baseline (figura 2.4.5) cu  $n=8$  intrări,  $m=8$  ieșiri și  $N=3$  nivele. Primul indicator urmărit este lățimea de bandă ( $\beta$ ), adică numărul de module de memorie comună ocupate dintre cele conectate pe ieșirile rețelei. Conform modelului propus, expresia de calcul a lui  $\beta$  este cea dată de relația (4.2.52), pe baza căreia am obținut graficul din figura 4.3.16, luând pentru fiecare procesor conectat pe intrări o rată a cererilor la memoria comună de  $\alpha=0,8$ . Pentru comutatoarele din interiorul fiecărei celule  $2 \times 2$  am admis o probabilitate de bună funcționare de  $p=0,999$ . În vederea verificării

credibilității modelului analitic, am elaborat și aici un program de simulare a funcționării sistemului (având algoritmul din Anexa 4), care ia în considerare așteptările cauzate de posibilitatea defectării comutatoarelor și conflictele datorate căilor comune. Rezultatele obținute prin simulare (pentru 50 de iterații), în ideea că o cerere refuzată din cauza unui conflict este menținută și în slotul următor, sunt prezentate pentru comparație pe același grafic cu cele calculate analitic. Se poate constata că cele două curbe sunt foarte apropiate, ceea ce întărește valabilitatea modelului matematic propus.

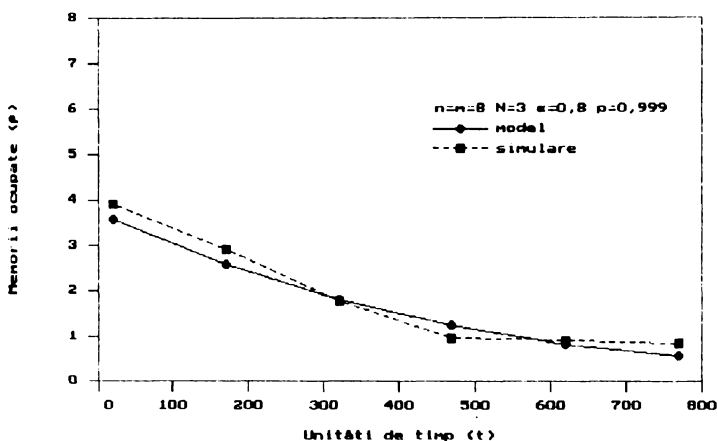


Figura 4.3.16.

Un alt indicator important de studiat este probabilitatea ca o cerere emisă de un procesor la o ieșire a rețelei să fie acceptată ( $P_A$ ). Această valoare este egală cu 1 numai dacă toate comutatoarele întâlnite pe calea respectivă sunt funcționale, calea este liberă și nu există nici o altă cerere simultană la memoria de pe ieșirea în cauză. Probabilitatea căutată se poate calcula cu relația (4.2.53), evoluția acesteia fiind reprezentată pe graficul din figura 4.3.17 pentru diferite fiabilități ale comutatoarelor. Se observă că probabilitatea de acceptare, care chiar pentru comutatoare perfecte ( $p=1$ ) este destul de mică ( $P_A \approx 0,6$ ), scade vertiginos pe măsura defectării comutatoarelor (pentru  $p \neq 1$ ). Acest lucru se explică prin faptul că între fiecare intrare și fiecare ieșire există câte o singură cale și defectarea unui comutator din cele patru considerate pe celulă (vezi figura 4.2.3) întrerupe imediat o cale pentru care nu există nici o alternativă. Este interesant de urmărit și modul în care este influențată probabilitatea de acceptare de modificarea ratei de acces ( $\alpha$ ) pentru un sistem dat. Acest lucru este reprezentat în figura 4.3.18, sistemele (cu 8, 16, respectiv 32 de intrări) fiind considerate aici perfect fiabile. De pe grafic rezultă că pe măsură ce dimensiunea rețelei ( $n$ ) este mai mare, probabilitatea de acceptare a cererii scade, acest lucru fiind mai pronunțat la rate de acces mai mari. De asemenea se poate vedea că  $P_A$  scade aproape liniar cu rata de acces, având valoarea minimă atunci când toate procesoarele emit cereri, această valoare fiind dată de:

$$P_{A_{\min}} = r_N \quad (4.3.11)$$

unde  $r_N$  se calculează cu relația de recurență

$$r_i = 1 - \left(1 - \frac{r_{i-1}}{2}\right)^2 \quad (4.3.12)$$

dedusă din (4.1.26) în care am luat  $\alpha=b=2$ , iar  $r_0=1$  și  $i = 1, 2, \dots, N$ . În particular pentru rețeaua cu  $N=3$  nivele, probabilitatea de acces minimă este  $P_{A_{\min}} = r_3 = 0,516$ .

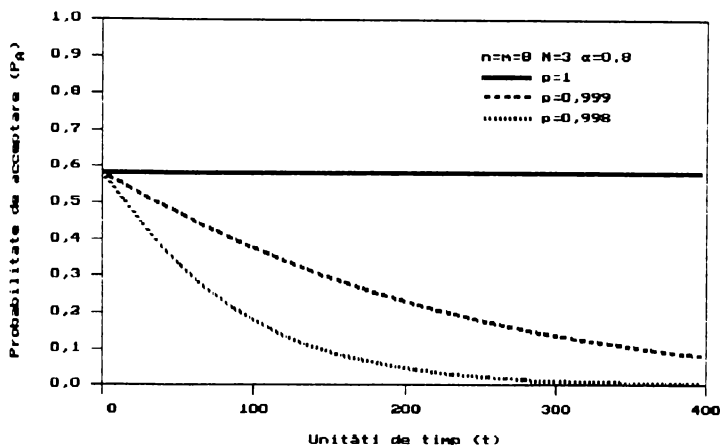


Figura 4.3.17.

În încheierea acestui capitol studiez comparativ  *timpul de răspuns ( $T_R$ )* pentru diferite tipuri de sisteme multiprocesor. Acest indicator reprezintă numărul mediu de unități de timp cât trebuie să aștepte un procesor până când cererea pe care a emis-o către o memorie comună este servită. Expresia de calcul a lui  $T_R$ , valabilă pentru toate sistemele tratate, a fost deja dată prin formula (4.2.27). În figura 4.3.19 am reprezentat grafic pe  $T_R$  pentru un sistem crossbar, un sistem cu magistrale multiple și o rețea multinivel, toate cu același număr de procesoare și memorii  $n=m=8$  și aceeași rată de fiabilitate a elementelor de interconectare ( $p=0,999$ ). Pentru sistemul cu magistrale am ales  $b=4$  magistrale, care reprezintă valoarea optimă pentru  $\alpha=0,6$  conform (4.3.8). De pe grafic se constată că timpul de răspuns crește pe măsură ce elementele de interconectare se defectează, dar această creștere este mult mai pronunțată la rețeaua MIN decât la celelalte două tipuri. Acest lucru se datorează faptului că distrugerea unui element de comutare din rețeaua multinivel duce la căderea în același timp a mai multor căi de comunicație. Este interesant de urmărit că evoluția curbelor pentru sistemele crossbar și cu magistrale este aproape identică. Acest lucru înseamnă, cel puțin prin prisma acestui indicator de performanță, că un sistem crossbar, cu căi separate și matrice de comutație complexă (aici cu 64 de comutatoare) nu poate fi considerat net superior unui sistem cu magistrale multiple (aici cu 4 magistrale). Explicația constă în faptul că defectarea unui comutator crossbar întrerupe complet comunicația între o pereche procesor-memorie, pe când căderea unei magistrale la sistemul celălalt nu întrerupe definitiv nici o astfel de legătură, urmarea fiind doar mărirea traficului pe magistralele rămase. Având în vedere acest avantaj, precum și faptul că numărul elementelor de interconectare este mai redus, îmi exprim opinia că

dintre cele trei alternative studiate un sistem cu magistrale multiple poate reprezenta o soluție economică și totuși performantă pentru implementarea unei aplicații multiprocesor.

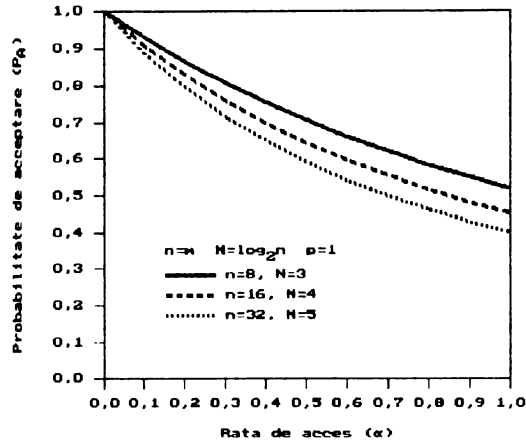


Figura 4.3.18.

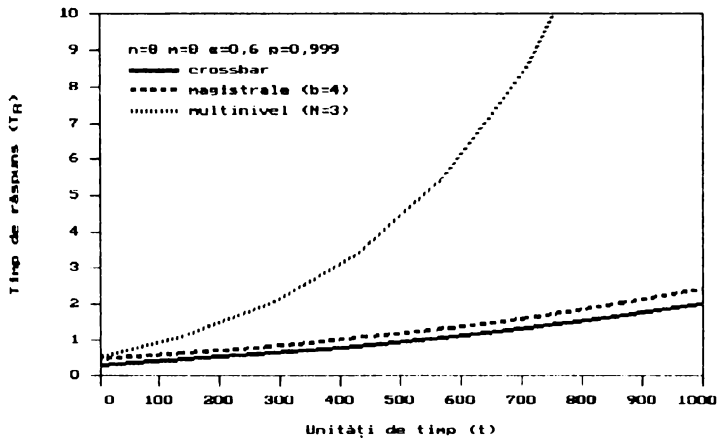


Figura 4.3.19.

Obiectul ultimelor două paragrafe l-a constituit prezentarea unui model matematic pentru studiul comportării diferitelor tipuri de sisteme multiprocesor, la care elementele de interconectare sunt expuse la defectare. Simularea pe calculator a confirmat valabilitatea modelului pentru indicatorii de performanță. Pe baza timpului mediu de răspuns, cea mai adecvată soluție s-a dovedit a fi sistemul cu magistrale multiple, având în vedere că acesta dispune și de o complexitate, respectiv cost mai redus ca celelalte.

## 5. FIABILITATEA REȚELOR DE CALCULATOARE

### 5.1. Caracteristici generale de fiabilitate și performabilitate

Analiza de fiabilitate a unui sistem distribuit de genul rețelilor de calculatoare utilizează o abordare teoretică cu grafuri, bazată pe modelarea rețelei cu un graf liniar în care nodurile corespund calculatoarelor, iar muchiile legăturilor de comunicație.

Fiabilitatea se referă la aptitudinea sistemului de a îndeplini funcția dorită. De aceea, primul lucru este de a identifica operațiile de rețea necesare, cea mai uzuală și naturală operație într-o rețea fiind comunicația. Înseamnă că în analiza și sinteza unei rețele de calculatoare, una din chestiunile fundamentale o reprezintă fiabilitatea și disponibilitatea căilor de comunicație între orice pereche de noduri. Aceste caracteristici sunt strâns legate pe lângă fiabilitatea și disponibilitatea componentelor individuale și de tehnicile de comunicație utilizate și de structura topologică a căilor de comunicație din sistem, justificându-se astfel studiul prin graful de sistem.

În analiza de fiabilitate se admit de obicei următoarele ipoteze asupra grafului de sistem [RAI90b]:

- toate elementele (noduri și muchii) sunt în modul activ (nu de rezervă)
- fiecare element poate fi reprezentat ca un dispozitiv bidirecțional
- starea oricărui element poate fi bună (operațională) sau defectă
- stările tuturor elementelor sunt statistic independente
- rețeaua este lipsită de cicluri orientate sau bucle, deoarece funcționarea sau căderea unei legături într-un ciclu orientat sau o buclă nu influențează fiabilitatea

Pe baza diferitelor definiții ale operațiilor acceptabile pentru funcționalitatea rețelei privind existența unor căi de comunicație între anumite perechi de noduri, au fost stabilite o mulțime de criterii de fiabilitate de tip *determinist* și *probabilist*.

Criteriile deterministe reprezintă o măsură a *vulnerabilității* rețelei de intercomunicație. Astfel ele dau o indicație asupra dificultății de a întrerupe funcționarea rețelei, prin determinarea numărului de noduri sau legături care trebuie să se defecteze pentru a duce la căderea rețelei.

Criteriile probabiliste de fiabilitate pun în evidență efectul defectărilor distribuite aleator în rețea. Astfel fiecărei legături și/sau nod  $i$  se asociază câte o probabilitate de defectare, ceea ce transformă graful sistemului într-un *graf de fiabilitate* (reliability graph). Pe baza acestui model, problema de analiză constă din determinarea probabilității ca rețeaua să fie operațională, în funcție de una din definițiile privind comunicarea dintre noduri.

**Criterii deterministe.** Sistemul de calcul se modelează printr-un graf liniar  $G = (V, E)$ , unde  $V$  este o mulțime nevidă de  $n$  noduri și  $E$  este o mulțime de  $b$  muchii. O linie de comunicație full-duplex între nodurile  $v_i$  și  $v_j$  corespunde muchiei neorientate  $[v_i, v_j]$ . Pe de altă parte, o legătură simplex de la  $v_i$  la  $v_j$  corespunde muchiei orientate  $(v_i, v_j)$ , acest caz fiind însă mai rar întâlnit. În ambele cazuri se spune că  $v_i$  și  $v_j$  sunt noduri

adiacente, iar muchia asociată este incidentă în aceste noduri. Se presupune de regulă că graful nu are muchii paralele și muchii de tipul  $[v_i, v_i]$ . La criteriile deterministe se evaluează parametri care nu țin cont de fiabilitățile elementelor componente, fiind specifici topologiei. Întrucât evaluarea acestor parametri se bazează pe o analiză a grafului asociat rețelei, prezintă în continuare câteva concepte de bază preluate din teoria grafurilor pe baza [MAYE72] și [WILK72].

Setul de noduri din  $G = (V, E)$  care sunt adiacente la  $v_i \in V$  se notează  $\Gamma(v_i)$ . Gradul nodului  $v_i$ , notat cu  $d_i = |\Gamma(v_i)|$ , reprezintă numărul de muchii incidente în  $v_i$ . Este ușor de observat că

$$\sum_{i=1}^n d_i = 2b \quad (5.1.1)$$

și dacă  $d = d_i$  pentru orice  $i$ , atunci  $d = 2b/n$  și  $G$  se numește un *graf omogen* sau *regulat* de gradul  $d$ . Două muchii de forma  $[v_i, v_j]$  și  $[v_j, v_k]$  sunt adiacente în  $G$ , deoarece au un nod comun. Secvența de noduri adiacente distincte  $[v_i, v_j, v_k, \dots, v_r, v_s, v_t]$  formează o cale între nodurile  $v_i$  și  $v_t$ , iar lungimea acestei căi este egală cu numărul de muchii situate pe ea. Oricare două căi între nodurile  $v_i$  și  $v_t$  se spune că sunt cu muchii disjuncte dacă nu au nici o muchie comună, respectiv cu noduri disjuncte dacă nu au nici un nod comun cu excepția lui  $v_i$  și  $v_t$ . Un circuit în graful  $G$  este o cale în care nodul inițial și final coincid, adică  $v_i = v_t$ . Lungimea circuitului este dată de numărul de muchii de pe calea închisă corespunzătoare. Lungimea minimă a tuturor circuitelor din  $G$  este numită *circumferința* grafului și se notează  $\tau(G)$ . Distanța  $l_{ij}$  dintre  $v_i$  și  $v_j$  este lungimea celei mai scurte căi dintre nodurile  $v_i$  și  $v_j$ . Lungimea maximă a tuturor căilor cu distanța cea mai scurtă din  $G$  se numește *diametrul* grafului și se notează  $k(G) = \max_{i,j} l_{ij}$ .

Un graf  $G = (V, E)$  este *conex* dacă există cel puțin o cale între oricare pereche de noduri  $v_i, v_j \in V$ . Un subgraf  $G_1$  al lui  $G$  este un graf ale cărui noduri sau muchii sunt conținute în  $G$ , adică  $G_1 = (V_1, E_1)$ , unde  $V_1 \subseteq V$  și  $E_1 \subseteq E$ . Dacă  $V_1 = V$ , atunci subgraful  $G_1$  se numește *subgraf de acoperire*. Un subgraf conex maxim din  $G$  se numește componentă a lui  $G$ . Un graf conex constă dintr-o singură componentă, pe când un graf care nu este conex trebuie să aibă evident cel puțin două componente. Un set (minim) de muchii a căror eliminare duce la creșterea numărului de componente cu unu se numește *tăietură* (minimală, primă) de muchii, iar un set (minim) de noduri cu aceeași proprietate este o *tăietură* (minimală, primă) de noduri. Când un nod este eliminat din graf, atunci sunt eliminate de asemenea toate muchiile incidente aceluia nod. O tăietură referitoare la o pereche specificată de noduri  $v_i$  și  $v_j$  dintr-un graf conex, numită și tăietură  $i-j$ , este aceea a cărei eliminare întrerupe toate căile dintre nodurile  $v_i$  și  $v_j$  (și ca rezultat  $v_i$  va face parte dintr-o componentă, iar  $v_j$  din cealaltă). Numărul minim de muchii din toate tăieturile  $i-j$ , notat cu  $C_{ij}^*$ , reprezintă conectivitatea de muchie (edge connectivity) dintre nodurile  $v_i$  și  $v_j$ . În mod similar, conectivitatea de nod (node connectivity)  $C_{ij}^n$  este dată de numărul minim de noduri din toate tăieturile  $i-j$ . Conectivitatea de muchie sau *coeziunea* grafului  $G$ , notată cu

$$C^*(G) = \min_{i,j} [C_{ij}^*(G)] \quad (5.1.2)$$

este numărul minim de muchii care trebuie eliminate pentru a întrerupe toate căile între o pereche de noduri din  $G$ . Un graf în care  $C^*(G) \geq m$  se numește *m-muchii conex*. Conectivitatea de nod sau simplu *conectivitatea* lui  $G$ , notată cu

$$C^n(G) = \min_{i,j} [C_{ij}^n(G)] \quad (5.1.3)$$

este egală cu numărul minim de noduri a căror eliminare deconectează graful. Dacă  $C^n(G) \geq m$ , atunci se spune că  $G$  este *m-conex*.

Într-un graf conex  $G$  cu  $n$  noduri și  $b$  muchii, un arbore  $T$  reprezintă subgraful de acoperire maxim (cu  $n-1$  muchii) care nu conține nici un circuit. Eliminarea oricărei muchii dintr-un arbore duce evident la deconectarea acestuia. De aceea se pot forma ușor tăieturi prime în  $G$  dintr-o muchie a arborelui și oricâte alte muchii ale lui  $G$  care nu fac parte din arbore. Toate tăieturile formate în acest fel corespund la vectorii binari cu  $b$  componente care sunt liniar independente și astfel dimensiunea spațiului tăieturilor prime este  $n-1$ . Prin adăugarea oricărei muchii la un arbore se formează un circuit. Circuitele formate din mai multe muchii ale arborelui și una din cele  $b-n+1$  muchii ale lui  $G-T$  corespund la vectorii binari cu  $b$  componente care sunt de asemenea liniar independenți. Astfel dimensiunea spațiului circuitelor lui  $G$  este  $b-n+1$ , care se mai numește nulitatea grafului, fiind notată cu  $\eta(G)$ .

Pornind de la modelul teoretic printr-un graf al rețelei de calculatoare, cele mai simple mărimi de fiabilitate deterministă sunt *coeziunea* (cohesion) și *conectivitatea* (connectivity) grafului. Ele sunt utilizate ca și o măsură a supraviețuirii sistemului, care se consideră a fi operațional dacă graful asociat este conex. În cazul rețelei de calculatoare conectivitatea de muchii, respectiv de noduri corespunde la numărul minim de legături de comunicație, respectiv de procesoare care trebuie să cadă pentru a întrerupe comunicația între orice pereche de calculatoare operative. Acest criteriu devine mai semnificativ dacă toate calculatoarele rețelei sunt la fel de importante. Altfel, pentru unele perechi de noduri ale grafului este de dorit ca să aibă o conectivitate de muchie sau de nod diferită de ale celorlalte.

Se poate arăta că numărul minim de noduri din orice tăietură  $i-j$  al unui graf este egal cu numărul maxim de căi cu noduri disjuncte dintre nodurile  $v_i$  și  $v_j$  (teorema lui Menger). În mod similar, conectivitatea de muchie dintre nodurile  $v_i$  și  $v_j$  este egală cu numărul maxim de căi cu muchii disjuncte dintre cele două noduri. Teorema fundamentală a lui Menger a simplificat determinarea conectivității de nod și de muchie dintre orice pereche de noduri din graf.

Inegalitatea lui Whitney arată că pentru orice graf  $G$  există relația:

$$C^c(G) \leq C^e(G) \leq d^*(G) \quad (5.1.4)$$

unde  $d^*(G) = \min(d)$ , adică este gradul minim al nodurilor din  $G$ . Inegalitatea nu poate fi îmbunătățită, deoarece oricare ar fi numerele întregi  $a$ ,  $b$  și  $c$  astfel încât  $a \leq b \leq c$ , există un graf  $G$  pentru care  $C^c(G) = a$ ,  $C^e(G) = b$  și  $d^*(G) = c$ . Se știe că în orice graf  $G$  cu  $b$  muchii și  $n$  noduri este adevărată relația (5.1.1) și gradul  $d$  mediu este  $2b/n$ . Deoarece  $d^*(G)$  nu poate fi mai mare decât gradul mediu al nodurilor,  $d^*(G) \leq 2b/n$ , unde „ $x$ ” semnifică cel mai mare întreg care nu depășește pe  $x$ , rezultă că:

$$C^c(G) \leq C^e(G) \leq d^*(G) \leq \lfloor 2b/n \rfloor \quad (5.1.5)$$

Relația dintre conectivitatea de nod, conectivitatea de muchie și gradul minim se datorează faptului că numărul maxim de căi cu noduri disjuncte dintre orice pereche de noduri  $v_i$  și  $v_j$  nu poate depăși numărul maxim de căi cu muchii disjuncte, care la rândul lui nu poate depăși minimumul dintre  $d_i$  și  $d_j$ . Ca urmare se poate defini un graf *conex maximal*, care corespunde unei rețele de calculatoare de fiabilitate maximă, ca fiind graful (neomogen) în care oricare ar fi  $i, j$ , există egalitatea:

$$C_i^c = C_i^e = \min(d_i, d_j) \quad (5.1.6)$$

Din proprietățile grafurilor  $m$ -conexe rezultă că un graf este 2-conex dacă și numai dacă oricare două noduri sunt așezate pe un circuit [WILK72]. În sinteza rețelilor 3-conexe se apelează la structura de tip roată (wheel)  $W_n$ , care constă dintr-un circuit cu  $n-1$  noduri și un nod în centru care este adiacent la fiecare dintre nodurile circuitului. Pe figura 5.1.1 se reprezintă grafurile  $W_7$ . Proiectarea rentabilă a unei rețele fiabile de calculatoare necesită realizarea unor grafuri  $m$ -conexe sau  $m$ -muchii conexe cu un număr minim de muchii sau noduri. În acest sens s-a definit grafurile cu *muchii critice* (edge critical) ca fiind un graf  $m$ -conex în care prin eliminarea oricărei muchii se obține un graf  $(m-1)$ -conex. O condiție necesară și suficientă pentru ca un graf  $m$ -conex să fie cu muchii critice este ca să nu existe mai mult de  $m$  căi cu noduri disjuncte între orice pereche de noduri adiacente. S-a arătat că un graf finit  $m$ -conex cu muchii critice conține mai mult de  $(m+1)^{1/2}/2$  noduri de grad  $m$ . Un graf  $m$ -conex este cu *noduri critice* (node critical) dacă în urma eliminării oricărui nod rezultă un graf  $(m-1)$ -conex. Într-un astfel de graf  $G$   $m$ -conex cu noduri critice,  $d^*(G) < (3m-1)/2$ . În mod similar un graf  $G$   $m$ -muchii conex se definește a fi cu noduri (sau muchii) critice, dacă ștergerea oricărui nod (sau muchie) din  $G$  duce la un graf  $(m-1)$ -conex. Nu există nici un graf  $m$ -muchii conex cu muchii critice pentru care  $d^*(G) > m$ . În cazul în care  $G$  are  $n$  noduri și  $d^*(G) \geq \lfloor n/2 \rfloor$ , atunci  $C^*(G) = d^*(G)$  și această limită dată de  $d^*(G)$  nu poate fi depășită.

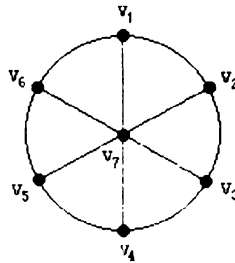


Figura 5.1.1.

Grafurile omogene (regulate) în care toate nodurile au gradul  $d$ , sunt maximal conexe dacă ele sunt  $d$ -conexe și  $d$ -muchii conexe. Pe baza celor expuse anterior, rezultă că un graf omogen conex maximal este cu muchii critice și cu noduri critice. Dacă toate nodurile dintr-o rețea de calculatoare sunt de aceeași importanță, o rețea de fiabilitate maximă corespunde unui graf omogen conex maximal. Aproape toate grafurile de acest fel sunt simetrice în raport cu nodurile lor (*grafuri cu noduri simetrice*), dispunând de proprietăți specifice [BOES86]. Rețeaua de fiabilitate maximă corespunde grafului (omogen) cu  $n$  noduri și  $b$  muchii, în care:

$$C^* = C^e = d^* = \lfloor 2b/n \rfloor \quad (5.1.7)$$

Un graf cu noduri simetrice  $G$  este hipoconex dacă  $C^*(G) < d(G)$ , unde  $d(G)$  este gradul unui nod din  $G$ . Caracterizarea acestor grafuri este dificilă, limita superioară minimă a raportului  $d(G)/C^*(G)$  fiind  $3/2$ . această limită însă nu poate fi atinsă. Însă orice graf cu noduri simetrice de gradul 2, 4 sau 6 este maximal conex, adică  $C^*(G) = d(G)$ . Pentru  $d(G) = 2$  grafurile devin un simplu poligon. Clasa grafurilor cu noduri simetrice care



au un număr prim de noduri constituie clasa poligoanelor în stea. Un graf în care nodurile  $v_i$  și  $v_j$  sunt adiacente constituie un poligon în stea dacă și numai dacă și nodurile  $v_{i+k}$  și  $v_{j+k}$  sunt de asemenea adiacente, unde  $1 \leq k \leq n-1$  și  $n$  reprezintă numărul de noduri din graf (indicii  $i+k$  și  $j+k$  sunt întregi modulo  $n$ ). În figura 5.1.2 se prezintă poligonul în stea de gradul 4 cu 7 noduri. Acest graf are 4 căi cu noduri și muchii disjuncte între fiecare pereche de noduri, fiind în consecință 4-conex. În general, poligoanele în stea reprezintă o clasă de grafuri cu noduri simetrice maximal conexe.

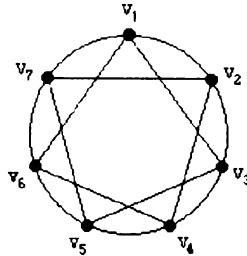


Figura 5.1.2.

Pentru a analiza fiabilitatea rețelelor de calculatoare din punct de vedere al conectivității de muchie și de nod, există diferite metode de determinare a acestor mărimi.

În teoria transmiterii informației, fiecărei muchii al unui graf orientat  $G$  se asociază un număr pozitiv care reprezintă *capacitatea* sa. Teorema lui Ford și Fulkerson afirmă că fluxul maxim dintre două noduri  $v_i$  și  $v_j$  este egal cu capacitatea tăieturii  $i$ - $j$  minime, unde capacitatea unei tăieturi este suma capacităților muchiilor din tăietură. Pe baza acestei teoreme s-a elaborat un algoritm pentru găsirea fluxului maxim dintre o pereche specificată de noduri ale unui graf orientat [IZVE94]. Pentru a putea aplica algoritmul la un graf neorientat  $G$ , acesta trebuie întâi transformat într-un graf simetric  $G'$  prin înlocuirea fiecărei muchii  $[v_i, v_j]$  din  $G$  cu o pereche de muchii  $(v_i, v_j)$  și  $(v_j, v_i)$  în  $G'$ . Capacitatea fiecăruia din aceste muchii orientate din  $G'$  este aceeași cu capacitatea originală a muchiei neorientate din  $G$ . Dacă toate muchiile din  $G$  au capacitatea unitară, atunci fluxul maxim dintre nodurile  $v_i$  și  $v_j$  în graful  $G$  reprezintă tocmai numărul de muchii din tăietura minimală  $i$ - $j$ , care este egal cu conectivitatea de muchie  $C_{ij}^e$  dintre nodurile  $v_i$  și  $v_j$ . Algoritm de calcul al lui  $C_{ij}^e$  pentru toate  $i, j$ , precum și al conectivității de muchie  $C^e(G)$  a grafului  $G$  necesită prin această metodă de etichetare un număr de  $n(n-1)/2$  iterații.

Conectivitatea de nod  $C_{ij}^n$  dintre nodurile  $v_i$  și  $v_j$  într-un graf orientat  $G'$  cu  $b$  muchii și  $n$  noduri poate fi determinată prin despicarea tuturor nodurilor lui  $G'$  și inserarea de noi muchii orientate între muchiile îndreptate spre fiecare nod, respectiv cele care ies din fiecare nod, urmate de aplicarea algoritmului de etichetare. Graful rezultat va avea  $2n$  noduri și  $b+n$  muchii, de aceea timpul de calcul este dublu decât în graful original. Conectivitatea de nod  $C_{ij}^n$  poate fi calculată și direct din graful orientat  $G'$ , iar având în vedere că fluxul rezultat prin aplicarea algoritmului de etichetare pentru  $C_{ij}^n$  poate fi folosit și la aflarea fluxului maxim dintre oricare alt nod  $v_i'$  și nodul  $v_j$  se reduce timpul de calcul. Pentru a verifica dacă un graf  $G$  neorientat cu  $n$  noduri este  $m$ -conex, nu trebuie

luate în considerare mai mult decât  $m(m-1)/2+m(n-m)$  din cele  $n(n-1)/2$  perechi de noduri. Acest lucru se bazează pe observația că pentru a verifica dacă  $G$  este  $m$ -conex este suficient a se verifica iterativ că în pasul  $j$  există  $m-j+1$  căi cu noduri disjuncte de la oricare nod  $v_j$  din  $G_j = G - \{v_1, v_2, \dots, v_{j-1}\}$  la toate celelalte noduri ale grafului  $G_j$ . Graful  $G_j$  este format din graful  $G$  prin eliminarea nodurilor  $v_1, v_2, \dots, v_{j-1}$  și a muchiilor aferente. Dacă deja s-a verificat existența a  $m$  căi disjuncte între  $v_i$  și fiecare din cele  $k$  noduri ale mulțimii  $V_j \subset \Gamma(v_j)$ , unde  $v_i \notin V_j$ , atunci pentru a verifica existența a  $m$  căi disjuncte între nodurile  $v_i$  și  $v_j$  este suficient a verifica existența a  $(m-k)$  căi disjuncte între  $v_i$  și  $v_j$  în graful  $G-V_j$ .

Un criteriu mai general de fiabilitate decât conectivitatea de muchie (coeziunea) unui graf este numărul minim de muchii care trebuie eliminate din graf pentru a izola un subgraf de  $m$  noduri de restul grafului. Această cantitate, numită *coeziunea generalizată*  $\alpha(m)$ , corespunde numărului minim de linii de comunicație a căror defectare va duce la izolarea unui set de  $m$  calculatoare de restul rețelei. Pe baza acestui criteriu, rețeaua de fiabilitate maximă corespunde evident grafului pentru care  $\alpha(m)$  are cea mai mare valoare posibilă pentru toate valorile lui  $m$ . Această mărime devine semnificativă pentru fiabilitatea unei rețele numai dacă probabilitatea de defectare a liniilor de comunicație este mult mai mare decât cea a nodurilor procesoare. Dacă însă este mai probabilă căderea procesoarelor decât a liniilor, atunci se poate folosi o mărime similară cu referire la noduri, notată  $\delta^n(m)$ .

Criteriile de mai sus sunt mărimi de fiabilitate mai generale decât conectivitatea de muchie sau de nod, deoarece fac posibilă distincția între diferite moduri de defectare ale sistemului. Este evident mult mai serioasă defectarea care duce la izolarea unei jumătăți a nodurilor din rețea de cealaltă jumătate, decât cea care izolează numai un nod de celelalte. (Ca urmare este de dorit ca o rețea fiabilă să aibă o valoare mult mai mare pentru  $\alpha(\lfloor n/2 \rfloor)$  decât pentru  $\alpha(1)$ ). Pe de altă parte, conectivitatea de muchie sau de nod reflectă numărul minim de muchii sau noduri defecte care izolează orice parte a rețelei de cealaltă, deci în orice graf cu conectivitatea de muchie  $C^*$  și cea de nod  $C^n$  există relațiile:

$$C^* = \min_m \{\delta(m)\} \quad (5.1.8)$$

$$C^n = \min_m \{\delta^n(m)\} \quad (5.1.9)$$

Pentru evaluarea lui  $\alpha(m)$  pentru un graf neorientat, dacă se notează un subset de  $m$  noduri cu  $M$ , complementul său cu  $L = N - M$  și numărul de muchii care interconectează cele două seturi cu  $\text{grd}(M)$ , atunci

$$\text{grd}(L) = \text{grd}(M) = \sum_{j \in M} \text{grd}(j) - 2b_M \quad (5.1.10)$$

iar

$$\delta(m) = \min_{\substack{M \\ |M|=m}} \{\text{grd}(M)\} = \delta(n - m) \quad (5.1.11)$$

unde  $b_M$  reprezintă numărul de muchii din subgraful care conține nodurile din  $M$ . În cazul grafurilor orientate,  $\text{grd}(M)$  se interpretează prin numărul de muchii orientate dinspre nodurile din  $M$  spre cele din  $L$ .

Limita inferioară a lui  $\delta(m)$  pentru un graf neorientat  $G$  cu  $n$  noduri pentru care  $d^*(G)$  reprezintă gradul minim al nodurilor, este dată de:

$$\delta(m) \geq m d^*(G) - m(m-1) \quad (5.1.12)$$

Această limită este atinsă pentru anumite valori numai dacă  $G$  conține  $m$  noduri, fiecare de grad  $d^*(G)$  și care formează un graf complet conex. În situația în care  $G$  este complet conex, toate nodurile au gradul  $n-1$ , limita inferioară fiind atinsă pentru toate valorile lui  $m$ , adică:

$$\delta(m) = m(n-m) \quad (5.1.13)$$

În cazul unui graf omogen neorientat  $G$  de gradul  $d$  și având  $n$  noduri,  $\delta(m)$  se poate calcula ușor ca fiind:

$$\delta(m) = md - 2 \max_{|M|=m} (b_M) \quad (5.1.14)$$

Un graf  $G$  se definește ca fiind  $\delta(m)$ -optimal, dacă pentru orice  $m$  conectivitatea generalizată  $\delta(m)$  este mai mare decât cea a oricărui alt graf omogen cu  $n$  noduri și având gradul  $d$ . S-a constatat că valoarea circumferinței grafului are o influență hotărâtoare asupra lui  $\delta(m)$ . Astfel, maximalizarea circumferinței într-un graf conex maximal este o condiție necesară dar nu și suficientă pentru ca graful să fie  $\delta(m)$ -optimal. Dintre grafurile omogene cu  $n$  noduri și grad  $d = n-n/k$ , clasa grafurilor complete  $k$ -partite este  $\delta(m)$  și  $\delta^n(m)$  optimală. Un graf complet  $k$ -partit este acela în care mulțimea nodurilor se poate partiționa uniform în  $k$  submulțimi disjuncte  $M_1, M_2, \dots, M_k$  și fiecare nod din submulțimea  $M_i$  este adiacent fiecăruia din celelalte  $n-n/k$  noduri ale grafului care nu fac parte din  $M_i$ . Un caz special îl reprezintă în această clasă grafurile complete bipartite, cu gradul  $n/2$  și circumferința 4 (grafuri de tip Singleton), care sunt grafuri omogene conexe maximale cu diametrul minim. În general, orice graf conex omogen cu  $n$  noduri având circumferința maximă  $t$  care satisface  $n/2 \leq t + \lfloor (t-1)/2 \rfloor$  este  $\delta(m)$ -optimal.

Utilizarea conectivității și a lui  $\delta(m)$  ca și mărimi de fiabilitate se bazează pe ipoteza că rețeaua este operațională în prezența defectelor dacă a rămas cel puțin câte o cale între fiecare pereche de noduri. O astfel de rețea poate însă să aibă un diametru excesiv de mare, care poate duce la întârzieri intolerabile în dirijarea mesajelor prin rețea. De aceea, o măsură mai elocventă a fiabilității ar putea fi numărul minim de noduri sau legături care trebuie să cadă pentru ca diametrul rețelei să crească peste o anumită valoare. Notând cu  $D^k(k, \omega)$  și  $D^n(k, \omega)$  numărul minim de muchii, respectiv noduri care trebuie eliminate din graful de diametru  $k$  pentru ca diametrul grafului rezultat să depășească pe  $\omega$ , atunci eliminarea oricărui  $D^k(k, \omega)-1$  muchii sau  $D^n(k, \omega)-1$  noduri duce la un graf cu diametrul mai mic sau egal cu  $\omega$ . Se presupune că un graf deconectat are diametrul  $\infty$ . Se poate arăta ușor că într-un graf cu  $n$  noduri și circumferința  $t$  care are conectivitatea de nod  $C^n$  și cea de muchie  $C^e$ , sunt valabile relațiile:

$$D^n(k, \omega) \leq C^n, \quad D^e(k, \omega) \leq C^e \quad (5.1.15)$$

și

$$D^n(k, t-2) \geq 2, \quad D^e(k, t-1) \geq 2 \quad (5.1.16)$$

În grafurile de tip Singleton și Moore, care au gradul  $d$ , diametrul  $k$  și circumferința  $t=2k$ , respectiv  $t=2k+1$ , relațiile devin:

$$D^d(k, t-2) = d, \quad D^d(k, t-1) = d \quad (5.1.17)$$

Atât coeziunea/conectivitatea generalizată, cât și diametrul grafului pot fi considerate deopotrivă mărimi de fiabilitate și performanță, deoarece reflectă într-o anumită măsură degradarea performanțelor din rețea (izolarea unor noduri, creșterea întârzierilor) în prezența defecțiunilor.

**Criterii probabiliste.** În prezența unor cauze distructive distribuite aleator prin rețea, măsurile deterministe nu sunt pe deplin adecvate pentru evaluarea fiabilității, o indicație mai precisă asupra fiabilității fiind dată de probabilitatea de a se întrerupe comunicația între anumite calculatoare, fiecărui nod și/sau muchie fiind asociate probabilități de defectare. Într-o astfel de rețea probabilistică oricare două noduri pot comunica dacă ambele sunt funcționale și există o cale de noduri și legături operative între ele. În aceste rețele se pot defini o mulțime de mărimi de fiabilitate, cele mai uzuale fiind următoarele [RAI90a]:

1. Fiabilitatea *sursă singulară la terminal* (SST, single source-to-terminal) reprezintă probabilitatea ca un anumit nod să poată comunica cu un alt nod specificat al rețelei. Aceasta se referă la situația când un calculator sursă intră în legătură cu un calculator destinație pentru accesul la un fișier de date.
2. Fiabilitatea *sursă multiplă la terminal* (MST, multisource-to-terminal) reprezintă probabilitatea ca un anumit număr de noduri sursă să poată comunica cu un nod terminal, adică un set de calculatoare să aibă acces la aceeași bază de date.
3. Fiabilitatea *sursă la terminal multiplu* (SMT, source-to-multiterminal) înseamnă probabilitatea ca un anumit nod sursă să comunice cu un anumit număr de noduri terminale operaționale. Pe această bază se pot modela o seamă de sisteme practice. De exemplu, se poate pune problema de a determina probabilitatea ca un anumit calculator să poată avea acces la fiecare din celelalte calculatoare ale rețelei.
4. Fiabilitatea *sursă multiplă la terminal multiplu* (MSMT, multisource-to-multiterminal) exprimată prin probabilitatea ca un anumit număr de noduri sursă să poată comunica cu un anumit număr de noduri terminale.
5. Fiabilitatea *globală a rețelei* (network reliability, overall reliability) reprezintă extinderea fiabilității MSMT la toate nodurile, adică probabilitatea ca toate nodurile să fie funcționale și să poată comunica între ele. În rețeaua de calculatoare aceasta se traduce prin probabilitatea ca fiecare calculator să aibă acces la toate celelalte.

Din această enumerare, primul indicator se mai numește *fiabilitate de tip terminal* (terminal reliability), iar celelalte sunt caracterizate prin termenul de *fiabilitate de tip multiterminal* (multiterminal reliability). Această din urmă noțiune poate fi interpretată în contextul cel mai general al fiabilității MSMT, celelalte tipuri fiind de fapt cazuri particulare ale acestuia. Metodele pentru calculul fiabilităților de tip terminal și

multiterminal au fost prezentate în capitolul 3, de aceea în continuare voi face câteva considerații privind impactul topologiei asupra indicatorilor de fiabilitate probabilistici, pe baza [WILK72].

Întrucât probabilitățile legăturilor/nodurilor nu sunt de regulă cunoscute înainte de implementare, se admite frecvent ipoteza simplificatoare a probabilităților egale, care permite comparația diferitelor topologii din punct de vedere al probabilităților de defectare. Fiabilitatea globală a rețelei reprezintă probabilitatea  $R_G$  ca graful  $G$  să rămână conex. Pentru un graf neorientat cu  $b$  muchii și  $n$  noduri, în care nodurile sunt perfect fiabile, iar muchiile au probabilități de defectare  $q_e$ , probabilitatea de conectare este:

$$R_G = \sum_{i=m}^b A_i (1 - q_e)^i q_e^{b-i} = 1 - \sum_{i=d}^b B_i q_e^i (1 - q_e)^{b-i} \quad (5.1.18)$$

unde  $A_i$  reprezintă numărul de subgrafuri de acoperire conexe ale lui  $G$  care au exact  $i$  muchii,  $B_i$  reprezintă numărul de subgrafuri de acoperire deconectate ale lui  $G$  care conțin exact  $b-i$  muchii,  $d$  este egală cu conectivitatea de muchie a grafului, iar  $m = n-1$  este numărul minim de muchii din oricare subgraf de acoperire conex. Pentru  $q_e \rightarrow 1$ ,  $R_G$  se poate aproxima cu  $A_{n-1}(1-q_e)^{n-1}$ , unde  $A_{n-1}$  este numărul de arbori din graf. Pentru  $q_e \rightarrow 0$ ,  $R_G$  este aproximativ  $1 - B_d q_e^d$ , unde  $B_d$  este numărul de tăieturi care conțin  $d$  muchii. Pe baza acestor relații Kelmans a arătat că există două grafuri  $G_1$  și  $G_2$ , astfel încât pentru valori mici ale lui  $q_e$  să existe  $R_{G_1} < R_{G_2}$ , iar pentru valori mari ale lui  $q_e$  să fie  $R_{G_1} > R_{G_2}$ . Rezultă că topologia unei rețele de fiabilitate maximă diferă în funcție de mărimea probabilității de defectare a muchiilor. Acest lucru complică enorm problema de sinteză a rețelelor pe baza criteriilor probabiliste de fiabilitate.

În cazul în care liniile de comunicație se consideră perfect fiabile și nodurile sunt expuse la defectare cu aceeași probabilitate  $q_v$ , probabilitatea de conectare  $R_G$  pentru graful cu  $n$  noduri care are o conectivitate de nod  $C^n = \theta$ , este

$$R_G = 1 - \sum_{i=\theta}^n N_i q_v^i (1 - q_v)^{n-i} \quad (5.1.19)$$

unde  $N_i$  reprezintă numărul de subgrafuri deconectate ale lui  $G$ , obținute în urma eliminării a exact  $i$  noduri. Dacă  $q_v \rightarrow 0$ , atunci  $R_G \cong 1 - N_\theta q_v^\theta$  și o rețea de fiabilitate maximă are evident o conectivitate de nod maximă. De aceea, pentru valori mici ale lui  $q_v$ , cea mai bună rețea de  $n$  noduri și  $b$  muchii are un număr minim de tăieturi de nod cu dimensiunea  $\lfloor 2b/n \rfloor$ . Pentru  $q_v$  apropiat de zero, graful bipartit complet cu  $n_1$  noduri într-o submulțime și  $n_2 > n_1$  noduri în cealaltă, are o probabilitate de conectare mai mare decât oricare alt graf de conectivitate  $n_1$  care nu are mai mult de  $n_1 n_2$  muchii.

În multe cercetări aptitudinea de supraviețuire (survivability) a rețelei se abordează sub forma fiabilității SST, adică a probabilității de a se păstra o conexinute între o pereche reprezentativă de terminale. Pentru un graf cu  $n$  noduri și  $b$  muchii, probabilitatea de a avea o comunicație între nodurile  $s$  și  $t$  este dată cu bună aproximație de relațiile

$$R_{st} \cong \sum_{i=0}^b A_{st}^i(i) (1 - q_e)^i q_e^{b-i} \quad (5.1.20)$$

pentru  $q_e \gg q_v$  și

$$R_{st} \cong \sum_{i=0}^{n-2} A_{st}^n(i) (1-q_v)^i q_v^{n-2-i} \quad (5.1.21)$$

pentru  $q_e \ll q_v$ . În aceste expresii,  $A_{st}^n(i)$  este numărul de combinații de  $i$  noduri astfel încât dacă ele sunt operaționale și restul de  $n-2-i$  noduri se defectează, atunci există cel puțin o cale de comunicație între  $s$  și  $t$ . Similar se interpretează și  $A_{st}^e(i)$ . Se poate calcula și probabilitatea de a nu se putea comunica între o pereche de noduri  $s$  și  $t$ , relațiile de calcul aproximativ fiind

$$F_{st} = 1 - R_{st} \cong \sum_{i=0}^b C_{st}^e(i) q_e^i (1-q_e)^{b-i} \quad (5.1.22)$$

pentru  $q_e \gg q_v$  și

$$F_{st} = 1 - R_{st} \cong \sum_{i=0}^{n-2} C_{st}^n(i) q_v^i (1-q_v)^{n-2-i} \quad (5.1.23)$$

pentru  $q_e \ll q_v$ . Coeficienții  $C_{st}^e(i)$  și  $C_{st}^n(i)$  reprezintă numărul de combinații de  $i$  muchii, respectiv noduri, astfel încât eliminarea numai a acestora din graf duce la distrugerea tuturor căilor dintre  $s$  și  $t$ . Ei sunt egali cu numărul total de tăieturi de muchie, respectiv de nod, de dimensiune  $i$  cu referire la nodurile  $s$  și  $t$ .

Pentru orice graf cu conectivitatea de nod și de muchie egală cu  $d$ ,  $C_{st}^e(i) = C_{st}^n(i) = 0$  pentru  $i < d$  și orice pereche de noduri  $s, t$ . De aceea, pentru  $1/2 > q_e \gg q_v$ , termenul dominant în expresia lui  $F_{st}$  este  $C_{st}^e(d) q_e^d (1-q_e)^{b-d}$ , iar pentru  $1/2 > q_v \gg q_e$ , acest termen este  $C_{st}^n(d) q_v^d (1-q_v)^{n-2-d}$ . Deoarece pentru valori mici (sub 0,5) ale lui  $q_e$  sau  $q_v$ , acești termeni se micșorează odată cu creșterea lui  $d$ , este evident că grafurile pentru rețele fiabile trebuie să fie maximal conexe. În mod similar, pentru  $q_e$  sau  $q_v$  mai mari de 0,5, termenul dominant în  $R_{st}$  este  $A_{st}^e(l) (1-q_e)^l q_e^{b-l}$  pentru  $q_e \gg q_v$ , respectiv  $A_{st}^n(r) (1-q_v)^r q_v^{n-2-r}$  pentru  $q_v \gg q_e$ , unde  $l$  reprezintă numărul de muchii din cea mai scurtă cale și  $A_{st}^e(l)$  este numărul de căi de lungime  $l$  dintre nodurile  $s$  și  $t$ . Numărul de noduri în cea mai scurtă cale este  $r=l-1$  și  $A_{st}^n(r) = A_{st}^e(l)$ . Rezultă că pentru valori mari ale lui  $q_e$ , termenul dominant în  $\min_{s,t} R_{st}$  este  $A_{st}^e(k) (1-q_e)^k q_e^{b-k}$ , unde  $k$  este limita

superioară a numărului de muchii de pe cea mai scurtă cale dintre oricare pereche de noduri, și aceasta este tocmai diametrul rețelei. Deoarece pentru orice rețea se încearcă maximizarea lui  $\min_{s,t} R_{st}$ , este necesar ca rețeaua să aibă un diametru minim dacă  $q_e$  sau  $q_v$

sunt mai mari de 0,5. Acest lucru pare să fie adevărat în condiții de trafic intens, în care  $q_e$  sau  $q_v$  desemnează probabilitatea ca o linie de comunicație sau un procesor să fie aproape de saturație. Observația este valabilă și în cazul particular în care  $q_v=0$  și probabilitatea de defectare a unei legături crește exponențial cu lungimea sa.

Dacă  $q_e$  și  $q_v$  sunt dați, atunci  $F_{st}$  variază numai în funcție de coeficienții  $C_{st}^e(i)$  și  $C_{st}^n(i)$ , care depind de topologia rețelei. Pentru sinteza unei rețele de fiabilitate ridicată, este necesar ca  $F_{st}$  să fie cât mai mic și  $R_{st}$  cât mai mare raportat la toate perechile de noduri. Înseamnă că mărimile  $\min_{s,t} R_{st}$  și  $\max_{s,t} F_{st}$  pot fi folosite ca măsuri probabiliste ale

fiabilității într-o rețea. Problema generală de sinteză a unei rețele optime constă în găsirea acelei configurații căreia îi corespunde o fiabilitate maximă la un cost minim [RAGH85] [HYUN75]. În majoritatea cazurilor se folosește metoda decompoziției bazată pe algoritmul "branch and bound", care în principiu este o cale de rezolvare a unei probleme de programare întregă prin enumerarea implicită a tuturor combinațiilor de soluții posibile [HWANG84][KUO87][CĂTU89b][JAN93].

În cadrul general al problemelor de fiabilitate se consideră și alte mărimi, cum ar fi gradul de serviciu de la nod la nod (reflectă capacitatea traficului între ele, care scăzând la zero poate duce la blocare de tip end-to-end blocking) sau timpul de transmitere între două noduri (end-to-end delay) [LIN78][CHAN80][GIRA83]. În multe situații fiecărei legături  $i$  se asociază un număr care reprezintă fluxul/capacitatea acesteia [LEE80] [RUEG86][KAKAS92]. Fiabilitatea se tratează sub aspectul integrat al existenței unei conexiuni între două noduri și a capacității de a se putea transmite un anumit flux de date între ele. Fluxul transmisiei pe o cale de la nodul  $s$  la nodul  $t$  este limitat de legătura cu capacitatea minimă de pe acea cale. Dacă există mai multe drumuri între cele două noduri, atunci după întreruperea unei legături comunicarea se poate menține, dar cu un alt flux maxim. Fiabilitatea între cele două noduri se va exprima ca și probabilitatea de a se putea transmite un anumit flux de date  $x_{st}$  între ele. Admițând o defecțiune singulară a legăturii  $e$  dintre  $s$  și  $t$  cu probabilitatea  $q_e$ , fiabilitatea transmisiei este dată de raportul dintre cantitatea totală de date transmisă cu defectarea legăturii  $e$  pentru un timp  $t_1$  și cantitatea totală de date de transmis într-un timp  $t$  ( $t_1 < t$ ):

$$\begin{aligned} R(x_{st}) &= \frac{tx_{st} - t_1[x_{st} - x_{st}(e)]}{tx_{st}} = 1 - \frac{t_1}{t} \left[ 1 - \frac{x_{st}(e)}{x_{st}} \right] = \\ &= 1 - q_e \left[ 1 - \frac{x_{st}(e)}{x_{st}} \right] \end{aligned} \quad (5.1.24)$$

unde  $x_{st}(e)$  este fluxul maxim transmis între  $s$  și  $t$  cu legătura  $e$  defectă. Abordarea reunită a fiabilității și capacității a condus la definirea unui indicator de performanță al rețelei sub forma unei fiabilități ponderate [RAI86b][AGGA88]:

$$R_{st} = \sum_i w_i \Pr\{\text{sistemul este în starea } S_i\} \quad (5.1.25)$$

unde  $S_i$  reprezintă o stare de reușită a sistemului (când există o cale  $s-t$ ). Relația 5.1.25 diferă de cea a fiabilității clasice  $R_{st} = \sum_i \Pr\{\text{sistemul este în starea } S_i\}$  prin ponderea  $w_i$ ,

care este raportul dintre capacitatea corespunzătoare stării  $S_i$  și cea maximă corespunzătoare stării cu toate legăturile funcționale.

Din punctul de vedere al prelucrării paralele și al distribuției taskurilor și fișierelor de date, fiabilitatea și performanțele pot fi privite sub aspectul resurselor operaționale necesare pentru execuția cu succes a unui anumit program.

Fiabilitatea sistemelor de prelucrare distribuită poate fi exprimată în funcție de fiabilitatea elementelor procesoare care execută programele, fiabilitatea elementelor procesoare care păstrează fișierele solicitate și fiabilitatea liniilor de comunicație folosite în transferul fișierelor de date. O măsură cantitativă asupra fiabilității unui sistem distribuit este *indicele de supraviețuire* (survivalability index), definit ca numărul preconizat de programe care rămân operaționale după căderea anumitor combinații de noduri și linii.

Această mărime nu este însă aplicabilă la sistemele mai mari, deoarece acestea pot avea multe stări și calculul devine complicat. De aceea în [PRAS86] se introduc alte două mărimi de fiabilitate (fiabilitatea programului distribuit și fiabilitatea sistemului distribuit), bazate pe observația că creșterea fiabilității și a toleranței la defecte se obține prin redundanța căilor de intercomunicație, precum și redundanța programelor și a fișierelor de date.

În modelul utilizat pentru un sistem de prelucrare distribuit, programele și fișierele de date sunt distribuite printre elementele procesoare care schimbă date prin liniile de comunicație. La execuția unui program cooperează mai multe procesoare, astfel încât programul care rulează într-un loc să poată solicita fișiere din altă parte. Pentru aceasta este necesar ca atât procesorul local, cât și cele care conțin fișierele solicitate, precum și comunicația între ele să fie funcționale. Funcționalitatea procesoarelor și a legăturilor fiind asigurată cu o anumită probabilitate, și execuția reușită a programului va fi posibilă cu o anumită probabilitate. Un exemplu este dat în figura 5.1.3, pe care *FA* desemnează fișierele disponibile (files available) într-un anumit nod, *FN*, reprezintă fișierele necesare (files needed) pentru execuția programului *PRG<sub>i</sub>*, iar *PRG* reprezintă programele care pot fi executate la nodul respectiv.

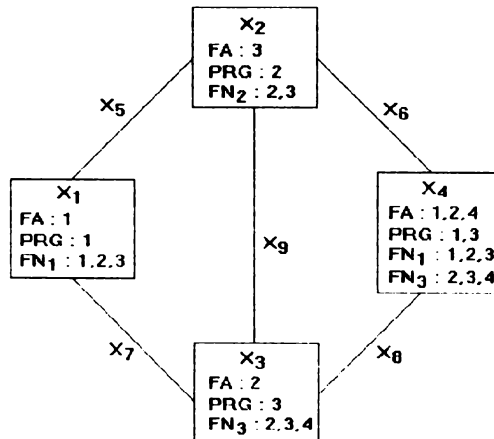


Figura 5.1.3.

*Fiabilitatea programului distribuit (DPR, distributed program reliability)* se definește ca probabilitatea de a se executa cu succes un anumit program distribuit, care rulează pe niște procesoare și necesită comunicarea cu alte procesoare pentru acces la fișiere aflate la distanță. Luând spre exemplu programul *PRG<sub>1</sub>*, execuția acestuia poate reuși dacă *x<sub>1</sub>* sau *x<sub>4</sub>* sunt funcționale și este posibil accesul la fișierele *F<sub>1</sub>*, *F<sub>2</sub>*, *F<sub>3</sub>*. Conexiunea de tip multiterminal corespunzătoare acestei condiții este:

$$C_m = [C_{1,2} \text{ \textasciitilde\textasciitilde\textasciitilde } (C_{1,4} \text{ SAU } C_{1,3})] \text{ SAU } C_{4,2}$$

Pentru calculul probabilității de îndeplinire se poate folosi metoda funcției boolene (paragraful 3.9), dar în [PRAS86] se indică și o altă posibilitate. Aceasta introduce



noțiunea de *arbore de acoperire a fișierelor* (FST, file spanning tree), ca fiind un arbore de acoperire care conectează nodul rădăcină (în care procesorul rulează programul considerat) la alte noduri, astfel încât să fie cuprinse toate fișierele necesare execuției programului. Din setul de arbori FST se rețin însă numai cei minimali (MFST, minimal FST), adică cei care nu sunt conținuți de un alt FST. Pentru execuția lui  $PRG_1$  în nodul  $x_1$  sau  $x_4$  din exemplul precedent, setul MFST este:

$$x_1x_2x_3x_5x_7, x_1x_2x_3x_5x_9, x_1x_2x_3x_7x_9, x_2x_4x_6, x_2x_3x_4x_8x_9$$

Oricare din acești arbori MFST poate duce la execuția cu succes a programului când toate elementele sale sunt funcționale. Fiabilitatea programului distribuit se va exprima ca probabilitatea de a avea cel puțin un MFST operațional, adică:

$$DPR(PR G_i) = \Pr\left(\bigcup_{j=1}^m MFST_j\right) = \Pr\left(\bigcup_{j=1}^m A_j\right) \quad (5.1.26)$$

unde s-a considerat că  $(A_1, \dots, A_m)$  reprezintă evenimentele corespunzătoare funcționării celor  $m$  arbori din setul MFST al lui  $PRG_i$ . Utilizând probabilități condiționate, relația se poate descompune în evenimente reciproc exclusive:

$$DPR(PR G_i) = \Pr(A_1) + \Pr(A_2) \cdot \Pr(\bar{A}_1|A_2) + \dots + \Pr(A_m) \Pr(\bar{A}_1 \cap \bar{A}_2 \cap \dots \cap \bar{A}_{m-1}|A_m) \quad (5.1.27)$$

unde  $\Pr(\bar{A}_1|A_2)$  înseamnă probabilitatea ca  $MFST_1$  să fie în stare neoperațională atunci când  $MFST_2$  este funcțional. În lucrarea amintită se reprezintă algoritmi pentru determinarea setului MFST (prin generarea sistematică a arborilor de acoperire traversând graful în maniera "breath-first search" de căutare în lățime, cu renunțarea la căutarea căilor între perechi de noduri) și evaluarea lui  $DPR$ . Lopez-Benitez combină algoritmul de evaluare a setului MFST cu un model bazat pe rețele Petri stohastice pentru calculul fiabilității  $DPR$ , care permite și luarea în considerare a reparațiilor globale [LOPEZ92].

*Fiabilitatea sistemului distribuit* ( $DSR$ , distributed system reliability) reprezintă o mărime globală a fiabilității sistemului, fiind exprimată ca probabilitatea de a avea toate programele executabile. Ea se poate scrie ca probabilitatea de intersecție a tuturor seturilor de arbori de acoperire minimali corespunzătoare programelor din sistem (considerate în număr de  $k$ ):

$$DSR = \Pr\left(\bigcap_{i=1}^k PRG_i\right) = \Pr\left(\bigcap_{i=1}^k MFST(PR G_i)\right) \quad (5.1.28)$$

Pe baza acestei relații s-a elaborat un algoritm de calcul prezentat în [PRAS86]. Expresiile pentru  $DPR$  și  $DSR$  pot fi folosite ca funcții obiect pentru rezolvarea unei probleme de optimizare în proiectarea unui sistem distribuit de înaltă fiabilitate și cost minim.

Alte două tehnici de determinare a fiabilității  $DPR$  sunt propuse în [KUMAR88a]. Ambele se bazează pe descompunerea matricii de conexiuni a rețelei, efectuată treptat, până când se ajunge la situații reciproc exclusive care satisfac condiția de conexiune formulată pentru execuția unui anumit program. Generarea arborilor prin descompunere urmează strategia "depth-first search" de căutare în adâncime. Primul algoritm este o

metodă în doi pași: prin descompunere se obține enumerarea conexiunilor multiterminal care duc la reușită, iar apoi se transpune expresia booleană a conexiunilor în funcția de fiabilitate prin oricare din procedeele cunoscute. Al doilea algoritm, numit FARE (Fast Algorithm for Reliability Evaluation), duce direct la generarea expresiei de fiabilitate, fără a fi necesară enumerarea conexiunilor multiterminal. Astfel se elimină dezavantajul de a opera cu un număr mare de evenimente corespunzătoare multiplelor posibilități de interconectare. Se definește și un *indice al costului de comunicație* (*CCI*, communication cost index) în funcție de timpul  $t$  de execuție al programului FARE pe un calculator dat, după relația:

$$CCI = \sum_{i=1}^p \sum_{k=1}^n \left[ PN_{ik} \times (FD_{ij} \times C_{jk})^t \right] \quad (5.1.29)$$

Parametrul *CCI* depinde de matricea programelor din sistem PN [ $p \times n$ ], matricea de distribuție a fișierelor FD [ $f \times n$ ] și matricea de conexiuni C [ $n \times n$ ], unde  $p$  reprezintă numărul total de programe,  $f$  numărul de fișiere și  $n$  numărul de noduri. *CCI* permite un studiu comparativ al timpului de execuție pentru diferite topologii și distribuții de fișiere, precum și pentru diverse metode de evaluare a fiabilității.

Algoritmul FARE se folosește de Kumar, Rai și Agrawal în [KUMAR88b] pentru determinarea fiabilității *DSR*, denumită acolo *fiabilitatea sistemului software* (*SSR*, software system reliability), dar având același înțeles: probabilitatea ca toate programele având fișiere de date distribuite să poată rula cu succes în ciuda unor defecte apărute la elemente procesare sau legături de interconectare. Mărimea *SSR* se poate interpreta ca o măsură a *robusteții software* (software robustness), adică a abilității sistemului de programe de a putea face față la defecte hard sau la erori de introducere. Ca sistemul să fie robust, apariția frecventă a acestor evenimente trebuie să aibă un efect minor asupra aplicației date [PĂUN93]. Ca exemple de sisteme unde se pune această problemă se pot aminti sistemele de tranzacție bancare sau ale agențiilor de turism, în care copii multiple ale fișierelor sunt distribuite prin rețeaua de calcul. Dezavantajul algoritmului FARE constă în inaplicabilitatea lui pentru programe distribuite care rulează pe mai multe noduri. Acest neajuns este eliminat la algoritmul FREA (Fast Reliability Evaluation Algorithm), bazat pe metoda factorizării care implică tehnici de reducere a grafului pentru calculul indicatorilor *DPR* și *DSR* [CHEN94].

Într-un sistem distribuit problema găsirii unei structuri optime de interconectare și a unei alocări optime a copiilor multiple ale programelor și fișierelor este de mare importanță. Alegând ca și criteriu de optim obținerea unei valori maxime pentru fiabilitatea distribuită a unui program, în condițiile unui număr minim de conexiuni și de copii, am formulat următoarele recomandări, care sunt de luat în considerare [KAKAS94b]:

a) Dacă un nod  $t$  are de făcut acces la nodul  $u$  sau  $v$  și dispune de câte o cale pentru acest lucru, atunci existența unei conexiuni directe între  $u$  și  $v$  nu duce la creșterea fiabilității, fiind inutilă. Într-adevăr, din figura 5.1.4.a se deduce că funcția booleană de conectivitate este

$$\Phi = \Phi_{t,u} \cup \Phi_{t,v} = x_1 \cup x_2 x_3 \cup x_2 \cup x_1 x_3 = x_1 \cup x_2 \quad (5.1.30)$$

și nu depinde de  $x_3$ .

b) Dacă nodul  $t$  are de făcut acces la nodurile  $u$  sau  $v$  și ele se găsesc pe aceeași cale, alocarea fișierului în nodul  $v$  nu duce la creșterea fiabilității, fiind inutilă. Pe figura 5.1.4.b se observă că

$$\Phi = \Phi_{t,u} \cup \Phi_{t,v} = x_1 \cup x_1x_2 = x_1 \quad (5.1.31)$$

nu depinde de  $x_2$ , fiind egală cu  $\Phi_{t,u}$ .

c) Dacă pentru execuția unui program nodul  $t$  are de făcut acces la nodul  $v$  și pe aceeași cale se află un alt nod  $u$  în aceeași situație cu  $t$ , atunci alocarea programului pentru  $t$  nu duce la creșterea fiabilității, fiind inutilă. De pe figura 5.1.4.c se deduce că

$$\Phi = \Phi_{t,u} \cup \Phi_{u,v} = x_1x_2 \cup x_2 = x_2 \quad (5.1.32)$$

nu depinde de  $x_1$ , fiind egală cu  $\Phi_{u,v}$ . Generalizarea acestui principiu este că dacă prin accesul dintr-un nod la resursele necesare se realizează evenimentul pentru execuția aceluiași program pe un alt nod, atunci alocarea pe nodul inițial nu are rost. Este cazul programului 3 pe nodul  $x_3$  în exemplul din figura 5.1.3.

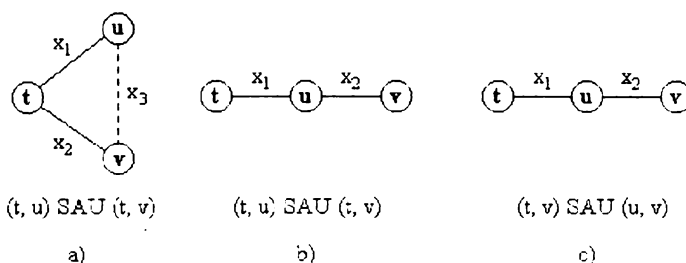


Figura 5.1.4.

Problema alocării mai multor copii ale fișierelor într-un sistem distribuit face obiectul multor cercetări. În [CHIU87] se prezintă un algoritm de optimizare a alocării unui număr limitat de copii a fișierelor la procesoare astfel încât să se minimizeze costul total de comunicație. Costul total de comunicație se determină prin însumarea costurilor tuturor căilor aferente pentru execuția programelor, în toate nodurile în care sunt definite. La alegerea fișierelor, dacă există mai multe copii disponibile, se va alege acela din nodul pentru care costul de comunicație este minim în raport cu nodul pe care se execută programul. La evaluarea costului se iau aici în considerare numai cele legate de comunicațiile de interogare (pentru obținerea de date), dar pe lângă acestea pot exista și alți factori într-un sistem, cum ar fi costurile de păstrare a fișierelor [IRANI82], de execuție [LEE92] sau cele de comunicație pentru reactualizarea fișierelor [TEWA92]. Costurile de comunicație între noduri depind de modul în care a fost proiectată rețeaua, iar numărul de copii din fiecare fișier ține de considerente financiare, de fiabilitate etc. Costul asociat unei legături poate semnifica întârzierea de propagare, lungimea, prețul liniei, combinațiile acestor mărimi sau chiar cantitatea de date transferate în timpul unei misiuni [SHATZ92]. Dacă se pune problema comunicației între toate nodurile rețelei astfel încât suma costurilor legăturilor implicate să fie minimă, atunci trebuie căutați

*arborii de acoperire minimali* (MST, minimal spanning tree) ai grafului asociat rețelei. Pentru aceasta există algoritmi care pornesc de la un nod oarecare sau algoritmi distribuți care pot fi executați de procesoarele din noduri în mod concurent și asincron [DALAL87].

Un arbore de acoperire minimal poate fi folosit pentru dirijarea difuzării informației (broadcast routing) într-o rețea de calculatoare care lucrează pe principiul comutării de pachete cu memorare și transmitere mai repede (store-and-forward packet-switching). O problemă mai generală este cea a dirijării informației de la un singur nod la un subset de noduri ale rețelei (multidestination routing) cu minimizarea cheltuielilor. Ca și criteriile de optim se pot considera costul de rețea NC (network cost), ca și suma costurilor aferente legăturilor implicate, sau costul de destinație DC (destination cost), ca fiind suma costurilor de pe căile cele mai scurte [BHAR83]. Pentru soluționarea acestor probleme de optim există o seamă de algoritmi, de regulă de tip euristic, bazați pe arborii MST, pe problema comerciantului călător, pe calea cea mai scurtă până la destinație. Dintre acestea, rezultatele cele mai bune pentru cazul cel mai defavorabil au fost date de algoritmul MST. În [SCHE84] se definește o difuzare optimă (optimal broadcast), ca fiind aceea căreia îi corespunde o lungime minimă, interpretată ca și interval de timp. Pentru rețele mari, datorită complexității calculului, se preferă algoritmi aproape de optim, de tip euristic. De asemenea, în astfel de sisteme un criteriu mai semnificativ este interpretarea fiabilității în sensul de a putea sau nu îndeplini sarcina cerută de utilizator.

**Performabilitatea.** În rețelele de calculatoare conectivitatea, ca o măsură a fiabilității, nu este întotdeauna edificatoare asupra funcționalității sistemului. Pe de o parte s-a constatat că defectele de conectivitate sunt relativ rare, iar pe de altă parte acestea nu afectează decât o parte din procesoare [RAI90a]. Astfel sistemul poate să rămână în continuare disponibil, dar cu performanțe reduse (graceful degradation), ceea ce necesită studierea unitară a fiabilității și performanțelor sistemului sub conceptul de performabilitate (performability).

În asemenea situații mărimile clasice de fiabilitate (funcția de fiabilitate, timpul mediu până la defectare, disponibilitatea sistemului) nu sunt suficiente pentru a lua în considerare diferențele nivele de performanță și capacitatea variabilă a sistemului de a executa anumite sarcini. Nivelul de performanță poate fi caracterizat de *capacitatea de prelucrare* (computation capacity)  $\alpha$ , într-o stare  $i$  a sistemului, care reprezintă numărul de unități de prelucrare utile (instrucțiuni, timp-CPU) pe unitatea de timp:

$$\alpha_i = \frac{T}{t} \quad (5.1.33)$$

Deoarece sistemele cu degradarea performanțelor posedă stări active care diferă între ele prin capacitatea de a executa sarcini și deoarece căderea sistemului apare numai când această capacitate scade sub o anumită valoare, se pot defini următoarele mărimi de fiabilitate legate de performanță [BEAU78]:

1. *Fiabilitatea de prelucrare* (computation reliability)  $R^*(t, T)$ , care este probabilitatea ca la momentul  $t$  sistemul să nu fie căzut și să execute corect un task de lungime  $T$  începând din momentul  $t$ .
2. *Prelucrarea medie înainte de defectare* (mean computation before failure)  $MCBF$ , care este valoarea așteptată a prelucrării înainte de prima cădere a sistemului.

3. *Pragurile de prelucrare* (computation thresholds)  $t_T$  și  $T_i$ , care înseamnă timpul după care fiabilitatea de prelucrare atinge o anumită valoare pentru un task de lungime  $T$ , respectiv lungimea maximă a taskului pe care îl poate executa sistemul începând cu momentul  $t$ .

4. *Disponibilitatea de prelucrare* (computation availability)  $a_c$ , care este valoarea așteptată a capacității de prelucrare la momentul  $t$  sau într-o stare staționară.

5. *Pragul de capacitate* (capacity threshold)  $t_c$ , care reprezintă timpul după care disponibilitatea de prelucrare atinge o anumită valoare.

Aceste mărimi se pot calcula similar cu mărimile clasice de fiabilitate, prin utilizarea modelelor Markov. Pentru aceasta lanțul Markov asociat sistemului se tratează în funcție de variabila de prelucrare  $T$  în loc de variabila timp  $t$ , efectuându-se substituția:

$$dt = dT / \alpha_i \quad (5.1.34)$$

Prin această tehnică reprezentarea din domeniul timpului se transformă în reprezentarea din domeniul prelucrării, așa cum se poate observa pe exemplul din figura 5.1.5. Aici s-a reprezentat un sistem cu  $N$  procesoare identice, cu ratele de defectare  $\lambda$  și capacitățile de prelucrare  $\alpha$  constante pentru fiecare procesor, fără posibilități de reparare. Evident în starea  $i$ , în care au mai rămas  $i$  unități funcționale,  $\lambda_i = i\lambda$  și  $\alpha_i = i\alpha$ . Probabilitatea de trecere din starea  $i-1$  în starea  $i$  în intervalul de timp  $dt$  este:

$$p_{i,i-1} = i\lambda c_i dt = i\lambda c_i dT / \alpha_i = i\lambda c_i dt / i\alpha = (\lambda c_i / \alpha) dT \quad (5.1.35)$$

unde  $c_i = c$  este acoperirea stării  $i$  (coverage), adică probabilitatea că sistemul se va reconfigura corect după defectarea unui procesor. Lanțul Markov transformat poate fi acum folosit pentru determinarea funcției de capacitate a diferitelor stări ale sistemului:

$$C_i(T) = 0 \quad \text{pentru } i \in \Psi$$

$$C_i(T) = \Pr\{\text{sistemul execută taskul de lungime } T \mid \text{prelucrarea a început în starea } i\} \quad \text{pentru } i \notin \Psi$$

unde  $\Psi$  este mulțimea stărilor în care sistemul este neoperațional. Această funcție, similară fiabilității din domeniul timp, se poate calcula rezolvând ecuațiile diferențiale ale modelului Markov cu necunoscutele  $P_i^*(T)$  (probabilitatea ca sistemul să fie în starea  $j$  după o prelucrare  $T$ ) considerând stările din  $\Psi$  absorbante (nu există nici o tranziție care pornește dintr-o stare defectă, în exemplul dat starea 0):

$$C_i(T) = \sum_{j \notin \Psi} P_j^*(T) \quad (5.1.36)$$

Fiabilitatea de prelucrare se calculează ca:

$$R^*(t, T) = \Pr\{\text{sistemul execută un task de lungime } T \mid \text{starea } = i \text{ la } t\} \cdot \Pr\{\text{starea } = i \text{ la } t \mid \text{starea } = I \text{ la } t=0\} = \sum_{i \notin \Psi} C_i(T) P_i(t) \quad (5.1.37)$$

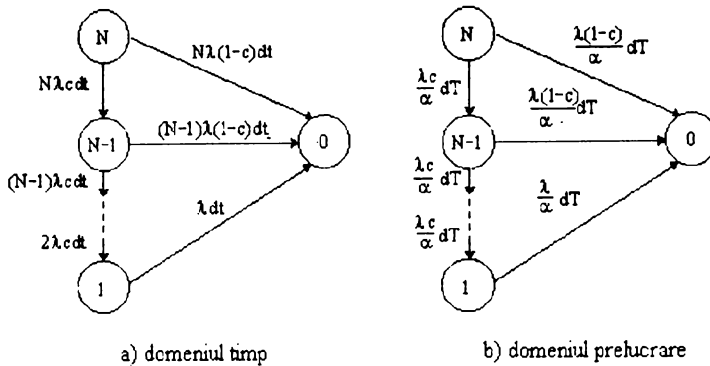


Figura 5.1.5.

Pe baza definiției se obțin și ceilalți indicatori de fiabilitate legați de performanță, care pentru  $\alpha = \text{const.}$  sunt dați de:

$$MCBF = \int_0^{\infty} C_I(T) dT = \alpha MTTF \quad (5.1.38)$$

$$a_c(t) = \alpha \sum_{i \in \Psi} P_i(t) \quad (5.1.39)$$

Dacă sistemul este în stare staționară, rezultă:

$$a_c = \alpha \sum_{i \in \Psi} P_i(\infty) = \alpha a_s \quad (5.1.40)$$

unde  $a_s$  este disponibilitatea clasică.

Făcând comparație între un sistem redundant cu două procesoare, unul de rezervă (standby) și un sistem cu degradare cu două procesoare (deci ambele active la început), se constată că fiabilitatea și  $MTTF$  sunt mai mari în primul caz, dar sistemul cu degradare dispune de o probabilitate mai mare de a executa sarcini lungi de prelucrare.

Dacă performanțele sistemului nu sunt degradabile (arhitecturi tolerante de tip "standby", redundanță  $N$ -modulară sau hibridă), fiabilitatea și performanța se pot trata separat în procesul de evaluare a *eficacității sistemului* (system effectiveness). Aceasta poate fi considerată în general ca mărimea așteptată a beneficiului (o valoare  $W_S$ ) adus utilizatorului de execuția taskurilor de către sistemul  $S$  în mediul său de lucru [MEYER80]:

$$Eff(S) = E[W_S] \quad (5.1.41)$$

În cazul sistemelor cu degradare însă, mărimile tradiționale de fiabilitate (probabilitatea succesului,  $MTTF$ ) nu sunt edificatoare, deoarece succesul poate avea diverse înțelesuri ca urmare a modificărilor structurale produse de defecțiuni. În acest caz valoarea  $W_S$  se exprimă ca o funcție de nivelul de performanță  $Y_S$  al sistemului, privit ca o

variabilă aleatoare, care ia valori într-un set de nivele de realizare  $A$  măsurabile (sesizabile) de utilizator:

$$W_S = w(Y_S) \quad (5.1.42)$$

unde  $Y_S \in A$ . Probabilitatea cu care sistemul  $S$  este capabil să-și îndeplinească funcția la nivelul  $a \in A$  din setul de realizări, este definită ca și *performabilitatea* lui  $S$  (performability):

$$P_S(a) = \Pr(Y_S = a) \quad (5.1.43)$$

Aceasta reprezintă o mărime unificată de performanță și fiabilitate, permițând evaluarea eficacității la nivelul sistemului:

$$E[w(Y_S)] = \sum_{a \in A} w(a)P_S(a) \quad (5.1.44)$$

unde  $w(a)$  se interpretează ca valoarea nivelului de performanță  $a$ .

O altă mărime care caracterizează din punctul de vedere al utilizatorului atât performanța, cât și fiabilitatea unui sistem distribuit este  *timpul mediu de răspuns* (average response time) în care sistemul rezolvă o cerere (execută un task), având în vedere căderile și reparațiile unor procesoare care pot să apară înainte de terminarea sarcinii. Evaluarea acestui timp se poate face pornind de la un model Markov al sistemului, în care se consideră că dacă două procesoare gestionează câte o copie a unei baze de date, atunci în cazul unei actualizări, modificările trebuie efectuate în ambele baze de date, pentru ca copiile să rămână identice [GARC85]. Întrucât oricare din procesoare poate să se defecteze, pe celălalt trebuie să se execute un algoritm de reconfigurare, în urma căruia cel funcțional va prelua și sarcina de a actualiza baza de date a celui defect. După reparație, pe ambele se va executa o nouă reconfigurare pentru a reveni la situația inițială. Pe baza acestui model se poate calcula timpul de răspuns așteptat al sistemului pentru îndeplinirea cererii remise la unul din procesoare, fiind necesară luarea în considerare a tuturor secvențelor posibile de stări care pot să apară, pornind de la o stare inițială. Condiția ca sarcina cerută să fie executată, este ca secvența să nu treacă prin nici o stare de avarie (evident ea poate trece prin stări de pauză când cererea este temporar suspendată, executându-se sarcini de reconfigurare). Pentru fiecare secvență se determină probabilitatea de apariție și durata preconizată, iar timpul cerut se obține ca o medie ponderată a timpilor preconizați pentru fiecare secvență. Datorită efortului de calcul, metoda se recomandă numai pentru sistemele mici, dar este foarte utilă atunci când se dorește compararea diferitelor strategii de reconfigurare.

Pe lângă timpul de răspuns, o altă mărime care poate face obiectul analizei este  *rata traficului* (traffic rate) datelor prin rețea (în biți / unitatea de timp sau blocuri / unitatea de timp), aplicabilă de exemplu la o rețea de comutare pe blocuri (burst switched network) [KUBAT86]. Aceasta se compune din mai multe comutatoare mici de linie dispuse în jurul unui comutator central de mare capacitate (hub), respectiv combinații de astfel de structuri, folosite pentru interconectarea calculatoarelor dintr-o rețea.

Considerând o rețea cu  $N$  comutatoare de linie, se pune problema de a determina pierderea în rata de transfer cauzată de defectarea unui element de comutare. (Se admit pentru simplitate numai defecte singulare și comutatorul central perfect fiabil, intenția fiind aici numai ilustrarea metodei). Există în total  $N$  stări diferite de defect, comportarea

rețelei fiind descrisă de un proces stohastic  $X(t)$ . În acest proces, starea sistemului oscilează între starea operațională (0) și o stare de defect ( $i$ ) cauzată de defectarea comutatorului  $i$  (figura 5.1.6). După deparare, procesul se reia totdeauna din starea 0. Se presupune că timpul  $U_i$  până la defectarea componentei  $i$  (defectul de tip  $i$ ) are o distribuție exponențială cu media  $1/\lambda_i$ ,  $i=1, 2, \dots, N$ , defectiunile fiind independente, iar timpul necesar pentru restabilire  $\Delta_i$  după producerea defectului de tip  $i$  are o distribuție cu media  $E[\Delta_i]$ . Se observă că procesul are o desfășurare ciclică, durata unui ciclu  $T$  fiind definită între două puncte de regenerare. Dacă  $T_0 = \min U_i$ , atunci timpul mediu petrecut în starea 0 este:

$$E[T_0] = E[\min U_i] = \int_0^{\infty} \left( \prod_{i=1}^N e^{-\lambda_i t} \right) dt = \lambda^{-1} \quad (5.1.45)$$

unde  $\lambda = \sum_{i=1}^N \lambda_i$ . Probabilitatea ca procesul  $X(t)$  să treacă din starea 0 în starea  $i$  în timpul unui ciclu este  $\lambda_i/\lambda$ . Durata preconizată a unui ciclu este:

$$E[T] = \lambda^{-1} + \lambda^{-1} \sum_{i=1}^N \rho_i \quad (5.1.46)$$

unde  $\rho_i = \lambda_i E[\Delta_i]$ . Valoarea așteptată a timpului  $T_i$  petrecută în starea  $i$  în timpul unui ciclu este:

$$E[T_i] = \frac{\lambda_i}{\lambda} E[\Delta_i] = \frac{\rho_i}{\lambda} \quad (5.1.47)$$

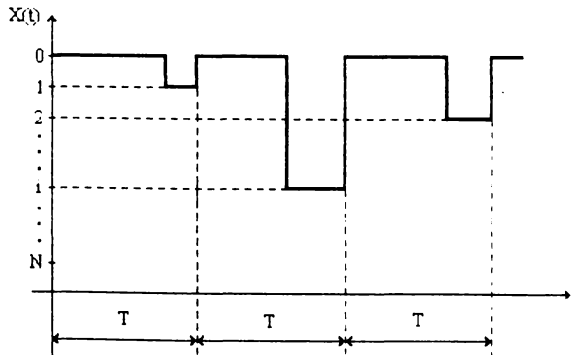


Figura 5.1.6.

Acum se poate calcula probabilitatea staționară ca procesul să fie în starea  $i$ :

$$p_i = \lim_{t \rightarrow \infty} \Pr\{X(t) = i\} = \frac{E[T_i]}{E[T]} = \rho_i p_0 \quad (5.1.48)$$



$$\text{unde } p_0 = \frac{1}{1 + \sum_{i=1}^N \rho_i}.$$

Dacă se notează cu  $r_i$  pierderea din rata de transfer cauzată de defectarea comutatorului  $i$ , aceasta are două componente:

- scăderea directă traficului cauzată de defectarea lui  $i$ , adică traficul care trebuia să pornească din  $i$ , respectiv care avea ca destinație nodul  $i$ :

$$W_i + \sum_{k=1}^N W_k g_{ki} \quad (k \neq i)$$

unde  $W_i$  este rata de transfer de intrare în nodul  $i$ , iar  $g_{ki}$  este fracțiunea din traficul originar din  $k$  care se termină în  $i$ .

- scăderea indirectă a traficului cauzată de căderea lui  $i$ , adică pierderea traficului de tranzit, scăderea datorată congestiilor, etc.:  $B_i$

Deci

$$r_i = B_i + W_i + \sum_{k=1}^N W_k g_{ki} \quad (k \neq i) \quad (5.1.49)$$

valorile pentru  $B_i$ ,  $W_i$ ,  $g_{ki}$  fiind determinate din matricea de dirijare aferentă sistemului. Având calculate mărimile  $p_i$  și  $r_i$  corespunzătoare fiecărui nod, se poate obține pierderea totală medie a ratei traficului la echilibru, ca o măsură unificată de performanță și fiabilitate  $AR$  (average reward):

$$AR = \frac{E[\text{pierdere pe un ciclu}]}{E[\text{durata ciclului}]} = \sum_{i=0}^N r_i p_i \quad (5.1.50)$$

Scăderea de performanță se poate exprima și prin eficiența traficului  $TE$  (traffic efficiency), o mărime adimensională:

$$TE = \frac{\text{trafic total} - \text{trafic pierdut}}{\text{trafic total}} = 1 - \frac{AR}{\text{trafic total}} \quad (5.1.51)$$

Criteriul prezentat este util în evaluarea comparativă a diferitelor structuri de rețea din punct de vedere al comportării la defecte singulare de cădere a unui nod.

O mulțime de modele integrate de performanță/fiabilitate se bazează pe interdependența dintre fluxul de programe și fiabilitatea sistemului de calcul. Fluxul de programe se consideră a fi o funcție de tipul sistemului și încărcarea acestuia, iar fiabilitatea depinde de structura sistemului și de procesele spontane de defectare care duc la erori sau căderi ale sistemului. Se consideră că o eroare duce la cădere numai dacă componenta în care a apărut eroarea este utilizată de programul care nu s-a executat corect. Căderile reduc debitul sistemului și necesită un proces de restabilire urmat de

reluarea programului. Astfel programele afectate de eroare au un timp mai mare de păstrare în memorie și gradul de utilizare al sistemului crește.

Pentru analiza influenței încărcării sistemului asupra defectelor și deci asupra performanțelor și fiabilității, există mai multe modele de *fiabilitate dependente de încărcare* (load-dependent reliability). Un astfel de model bazat pe rețele de cozi de așteptare (queueing networks) de tip BCMP [AJMO86] este prezentat în [SCHO86]. Rețeaua BCMP constă din  $N$  noduri (centre de deservire) și un număr de taskuri grupate pe  $R$  clase. Un task din clasa  $r$  care își termină serviciul în nodul  $i$ , va cere să fie servit în nodul  $j$  din clasa  $s$  cu o probabilitate fixă de tranziție  $p_{r,r,s}$ . Fluxul taskurilor în acest spațiu de noduri și clase poate fi descompus în sublanțuri, fiecare dintre ele descriind un anumit "job" care se poate executa pe sistemul modelat. După regula de ordonare pot exista patru tipuri de noduri: primul-venit-primul-servit, cu divizarea timpului, cu servire continuă (fără așteptare) și ultimul-venit-primul-servit. Încărcarea poate fi caracterizată de rata de sosire  $\lambda_{\text{job}}$  a fiecărui program și de timpul său de stocare. Pentru fiecare program se poate calcula *riscul aferent* (risk per job):

$$P_{\text{job}} = \Pr\{\text{job afectat}\} = \frac{\sum \rho_i}{\lambda_{\text{job}}} \quad (5.1.52)$$

unde  $\sum \rho_i$  este rata de defectare totală pentru fluxul de taskuri corespunzător programului. Riscul unui program de a suferi un defect crește cu încărcarea sistemului, deoarece ratele de defectare rezultante cresc mai mult decât proporțional cu încărcarea.

Un alt mod de evaluare al eficacității unui sistem distribuit, care combină performanța și fiabilitatea, este cel bazat pe modelul Markov cu câștig (MRM, Markov reward model) [SMITH88]. Comportarea sistemului se descrie printr-un lanț Markov continuu în timp și fiecărei stări  $i$  se asociază o rată de câștig ca și o măsură a performanței. Lanțul reflectă procesul structural al evoluției sistemului prin diferite stări, iar cu câștigurile aferente se obține structura de performanță. Libertatea de a putea modifica atât procesul structural al stărilor, cât și structura de performanțe, permite modelarea unei mari varietăți de situații. Astfel, la o extremă se situează modelul pur de performanță care ignoră defectele și reparațiile, considerând că lupta pentru suficiență memorie determină capacitatea sistemului de a-și îndeplini sarcinile. La cealaltă extremă, modelul pur de disponibilitate ignoră diferitele nivele de performanță, în afară de cea operațională și defectă. Evoluția în timp a sistemului este caracterizată de un proces stohastic  $\{Z(t), t \geq 0\}$  cu stări finite  $S = \{1, 2, \dots, M\}$ , astfel că  $Z(t) \in S$  reprezintă starea sistemului la momentul  $t$ . Dacă fiecărei stări  $i$  i se asociază un nivel de performanță  $r_i$ , atunci se poate construi vectorul de performanță (câștig) al sistemului  $R = [r_i]$ . Nivelul de performanță la momentul  $t$  este dat de valoarea variabilei aleatoare  $X(t)$ :

$$X(t) = r_{Z(t)} \quad (5.1.53)$$

Valoarea preconizată a lui  $X(t)$ , și anume  $E[X(t)]$  reprezintă performanța așteptată de la sistem în momentul  $t$  și este tocmai *disponibilitatea de prelucrare* (computation availability), care se poate calcula ca:

$$E[X(t)] = \sum_i r_i P_i(t) \quad (5.1.54)$$

unde  $P_i$  reprezintă probabilitatea ca sistemul să fie în starea  $i$  la momentul  $t$ , adică  $P_i(t) = \Pr\{Z(t) = i\}$ .

Un MRM în care stărilor operaționale li se asociază rata de câștig 1, iar celor neoperaționale 0, constituie un model de disponibilitate (availability model). De multe ori este convenabilă determinarea disponibilității în exploatare a unui sistem dat. Astfel în [SPRA81] se prezintă un astfel de exemplu, bazat pe urmărirea comportării unui sistem de comunicații cu ocazia a peste 400 de transmisii. Datele culese permit interpretarea efectului diferiților parametri (viteza de transmisie, distanța, durata întreruperii) asupra disponibilității sistemului.

Conceptul de performabilitate este destul de general și, așa cum s-a văzut, în funcție de modelul ales, respectiv variabila considerată pentru caracterizarea nivelului de performanță poate fi specificată prin mărimi stricte de performanță, de fiabilitate sau care fac referire la fiabilitate cum ar fi disponibilitatea.

## 5.2. Modelul de performabilitate propus

Condiția de funcționalitate a unei rețele de calculatoare, care stă la baza determinării indicatorilor de fiabilitate, constă în existența unei conectivități între anumite perechi de noduri ale grafului asociat rețelei. Acest criteriu limitează însă studiul performanțelor sistemului doar la defectarea unor legături/noduri, fără să țină seama și de o altă caracteristică a comportării reale a sistemului, și anume traficul din rețea. Încărcarea cu trafic a liniilor are, în plus față de condiția de conectivitate, o influență hotărâtoare asupra unui transfer reușit de date între două noduri.

În figura 5.2.1.a consider două noduri ale unei rețele interconectate printr-o legătură  $e$  care permite transferul de date de la  $s$  la  $t$ . Pentru analiza în timp a transmisiei nu este adecvat un model discret ca cel propus la sistemele multiprocesor strâns cuplate din capitolul precedent, deoarece în cazul rețelelor, așa cum am arătat la paragraful 2.1, transferul se realizează uzual de la nod la nod sub formă de pachete de date. Ca urmare se impune luarea în considerare a unei rate de transmisie, respectiv de prelucrare a pachetelor într-un spațiu continuu al timpului. Datorită proprietăților favorabile pe care le are (vezi paragraful 3.4), voi admite că atât transmisia, cât și prelucrarea se supun unei legi de distribuție negativ exponențială, cu rata constantă. Dacă pentru exemplul dat în figură nodul sursă  $s$  transmite pachete către cel destinație cu o rată constantă  $\lambda$ , iar acesta le prelucerează cu o rată constantă  $\mu$ , între ele se formează o coadă de așteptare, reprezentată fizic de un buffer de ieșire în care calculatorul sursă depune pachetele de transmis (figura 5.2.1.b). Serverul aferent cozii este reprezentat de calculatorul destinație. Condiția de stabilitate pentru ca lungimea cozii să nu crească la infinit, este ca rata medie de transmisie să fie mai mică decât rata medie de servire ( $\lambda < \mu$ ). Intensitatea traficului corespunzătoare acestei cozii (de tip M/M/1 [JAIN91]) și implicit liniei de transmisie, este:

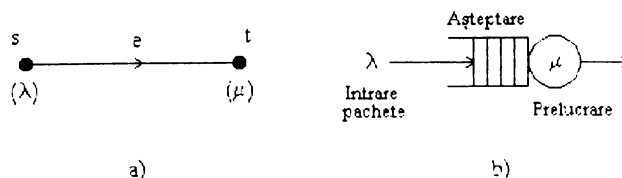


Figura 5.2.1.

$$\rho = \frac{\lambda}{\mu} \quad (5.2.1)$$

Sistemul de așteptare considerat poate fi modelat printr-un lanț Markov cu stări discrete (figura 5.2.2). Starea sistemului este reprezentată de numărul de pachete ( $n$ ) din sistem. Sosirea în coadă a unui nou pachet provoacă schimbarea stării cu  $+1$ , iar părăsirea stației după servire duce la modificarea stării cu  $-1$ . Un astfel de proces Markov în care tranzițiile se limitează numai la stările vecine se numește proces de naștere și moarte (birth-death process).

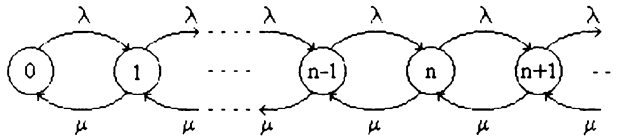


Figura 5.2.2.

Pentru a calcula probabilitatea staționară  $p_n$  ca procesul să fie în starea  $n$ , presupunem că atât timpul între două sosiri, cât și timpul de servire au o distribuție exponențială. Dacă  $S(t)$  reprezintă numărul de pachete din sistem la momentul  $t$ , atunci trecerea sistemului din starea  $n$  în starea  $n+1$ , respectiv  $n-1$  în intervalul de timp foarte scurt  $\Delta t$  este descris de următoarele probabilități:

$$\begin{aligned} \Pr\{S(t+\Delta t) = n+1 \mid S(t) = n\} &= \lambda \Delta t \\ \Pr\{S(t+\Delta t) = n-1 \mid S(t) = n\} &= \mu \Delta t \end{aligned}$$

Dacă nu există nici o sosire sau plecare în intervalul  $(t, t+\Delta t)$ , atunci sistemul va rămâne în starea  $n$  cu probabilitatea:

$$\Pr\{S(t+\Delta t) = n \mid S(t) = n\} = 1 - \lambda \Delta t - \mu \Delta t$$

Notând cu  $p_n(t)$  probabilitatea ca sistemul să fie în starea  $n$  la momentul  $t$ , se poate scrie următorul sistem de ecuații:

$$\begin{aligned} p_0(t+\Delta t) &= (1-\lambda \Delta t) p_0(t) + \mu p_1(t) \\ p_n(t+\Delta t) &= \lambda \Delta t p_{n-1}(t) + (1-\lambda \Delta t - \mu \Delta t) p_n(t) + \mu \Delta t p_{n+1}(t) \quad (n \geq 1) \end{aligned} \quad (5.2.2)$$

Trecând la limită, ultima ecuație se mai poate scrie:

$$\frac{dp_n(t)}{dt} = \lim_{\Delta t \rightarrow 0} \frac{p_n(t+\Delta t) - p_n(t)}{\Delta t} = \lambda p_{n-1}(t) - (\lambda + \mu) p_n(t) + \mu p_{n+1}(t) \quad (5.2.3)$$

Având în vedere faptul că în stare staționară  $\lim_{t \rightarrow \infty} p_n(t) = p_n$  și  $\lim_{t \rightarrow \infty} \frac{dp_n(t)}{dt} = 0$ , sistemul de ecuații (5.2.2) devine:

$$\begin{aligned} -\lambda p_0 + \mu p_1 &= 0 \\ \lambda p_{n-1} - (\lambda + \mu) p_n + \mu p_{n+1} &= 0 \quad (n \geq 1) \end{aligned} \quad (5.2.4)$$

Dacă se face în ultima ecuație substituția  $Z_n = -\lambda p_n + \mu p_{n+1}$  se obține  $Z_n = Z_{n-1}$ , iar din prima ecuație a sistemului (5.2.4) rezultă  $Z_0 = 0$ . Ca urmare, în cazul staționar  $Z_n = 0$  ( $n \geq 0$ ), adică:

$$p_{n+1} = \frac{\lambda}{\mu} p_n \quad (n \geq 0) \quad (5.2.5)$$

Prin recurență se deduce imediat expresia pentru probabilitatea de a avea  $n$  pachete în sistem:

$$p_n = \left( \frac{\lambda}{\mu} \right)^n p_0 = \rho^n p_0 \quad (5.2.6)$$

Deoarece suma tuturor probabilităților trebuie să fie 1, probabilitatea de a avea zero pachete în sistem se poate calcula ca:

$$p_0 = \frac{1}{1 + \sum_{n=1}^{\infty} \rho^n} = 1 - \rho \quad (5.2.7)$$

Substituind pe  $p_0$  în expresia lui  $p_n$  se obține

$$p_n = (1 - \rho) \rho^n \quad (n = 0, 1, 2, \dots) \quad (5.2.8)$$

ceea ce arată că numărul de pachete din sistem are o distribuție geometrică. Ca urmare, numărul mediu de pachete în sistemul format din coadă + stație este dat de

$$E[n] = \sum_{n=1}^{\infty} n p_n = \sum_{n=1}^{\infty} n (1 - \rho) \rho^n = \frac{\rho}{1 - \rho} \quad (5.2.9)$$

iar numărul mediu de pachete aflate în coada de așteptare este:

$$E[n_q] = \sum_{n=1}^{\infty} (n-1) p_n = \sum_{n=1}^{\infty} (n-1) (1 - \rho) \rho^n = \frac{\rho^2}{1 - \rho} \quad (5.2.10)$$

Probabilitatea ca în coada de așteptare să nu fie nici un pachet, adică sistemul să fie în starea 0 sau 1, este dată de:

$$p^0 = \Pr(n_q = 0) = p_0 + p_1 = 1 - \rho^2 \quad (5.2.11)$$

Probabilitatea ca în coadă să se afle  $m$  pachete ( $m \geq 1$ ) este egală cu probabilitatea ca în sistem să fie  $m+1$  pachete, deci:

$$\Pr(n_q = m) = (1 - \rho) \rho^{m+1} \quad (m \geq 1) \quad (5.2.12)$$

Menționez că probabilitățile de mai sus pot fi determinate și în cazul netaționar, urmând procedeul de calcul din [МНОС73].

Într-o rețea de calculatoare cu  $M$  linii de comunicație bidirecționale pot fi considerate  $2M$  cozi în care se înșiruiesc pachetele de date transmise între noduri, ca în figura 5.2.3 unde  $M=3$ . Pentru analiza intensității de trafic prin fiecare linie, construiesc un model care are la bază următoarele presupuneri:

- fiecărui nod  $v_i$  i se asociază o rată  $\lambda_i = \text{const.}$  cu care emite pachete către celelalte noduri, selectate în mod uniform
- pachetele transmise de un nod sunt distribuite uniform pe liniile de ieșire ale acestuia
- fiecare nod  $v_i$  constituie o stație cu rata de servire  $\mu_i = \text{const.}$  care prelucrează pachetele sosite în cozile aferente liniilor de intrare în acel nod
- o fracțiune  $\alpha < 1$  a pachetelor sosite la un nod sunt reținute de acesta în urma prelucrării (care se consideră a fi destinate acestui nod), iar restul sunt transmise mai departe în rețea pe liniile de ieșire

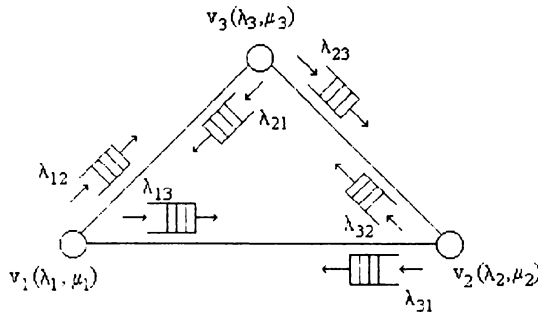


Figura 5.2.3.

În cazul cel mai general al unui nod  $v_i$  cu  $k$  linii de intrare și  $l$  linii de ieșire vor exista  $k$  cozi de așteptare în care sosesc pachete cu ratele  $\lambda_1, \lambda_2, \dots, \lambda_k$  și care vor fi prelucrate succesiv de serverul aferent nodului cu rata de prelucrare  $\mu$  (figura 5.2.4). Timpul între două sosiri consecutive pe o latură având o distribuție exponențială, procesul corespunzător este de tip Poisson. În acest caz este adevărată proprietatea de aditivitate a ratelor medii la reunirea șirurilor Poisson și rata medie totală a pachetelor care trebuie prelucrate în nodul  $v_i$  este:

$$\lambda_{tot_i} = \sum_{j=1}^k \lambda_j \quad (5.2.13)$$

Ținând cont de presupunerile făcute mai sus, rata medie de transmitere a pachetelor pe o latură de ieșire, respectiv numărul mediu de pachete în fiecare coadă se pot exprima în felul următor:

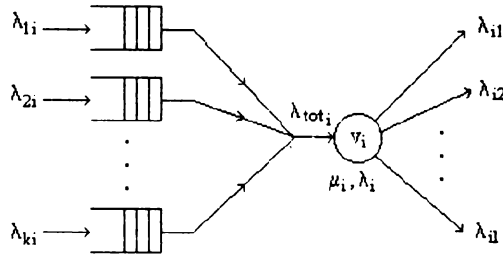


Figura 5.2.4.

$$\lambda_{ij} = \frac{(1-\alpha)\lambda_{toti} + \lambda_i}{l} \quad (j = 1, 2, \dots, l) \quad (5.2.14)$$

$$E[n_{q_j}] = \frac{\lambda_{ji}}{\lambda_{toti}} \frac{\rho_{ji}^2}{1 - \rho_{ji}} \quad (j = 1, 2, \dots, k) \quad (5.2.15)$$

unde  $\rho_i = \frac{\lambda_{toti}}{\mu}$ . Probabilitatea ca în situația în care sistemul de  $k$  cozi aferente lui  $v_i$  se află  $m$  pachete ( $m \geq 1$ ), dintre care în coada  $ji$  ( $j = 1, 2, \dots, k$ ) să se afle  $m_j$  pachete, se calculează cu ajutorul probabilităților condiționate, și anume:

$$p_{ji}(m, m_j) = \Pr(m_j | m) \cdot \Pr(n_q = m) \quad (5.2.16)$$

unde  $\Pr(m_j | m)$  reprezintă probabilitatea lui  $m_j$  în coadă condiționată de existența lui  $n_q = m$  în sistemul de cozi. Acesta se poate exprima admitând că sistemul din figura 5.2.4 este similar cu un model cu o singură stație, sosiri după legea multiplă Poisson și timp de servire exponențial [МИНОС67], și atunci pentru o servire în ordinea sosirilor:

$$\Pr(m_j | m) = \binom{m}{m_j} \left( \frac{\lambda_{ji}}{\lambda_{toti}} \right)^{m_j} \left( 1 - \frac{\lambda_{ji}}{\lambda_{toti}} \right)^{m-m_j} \quad (5.2.17)$$

Al doilea factor al relației (5.2.16) corespunde lui (5.2.12), în care se folosește  $\rho$  în loc de  $\rho$ .

Întrucât am considerat o distribuție uniformă a pachetelor printre ieșirile dintr-un nod  $v_i$ , rata medie de emiteră a pachetelor pe fiecare din aceste laturi va fi  $A_i = \lambda_{i1} = \lambda_{i2} = \dots = \lambda_{il}$ . Pentru rețeaua compusă din  $N$  noduri mărimile  $A_i$  ( $i = 1, 2, \dots, N$ ) reprezintă necunoscutele care trebuie determinate pentru a putea caracteriza complet intensitatea traficului și numărul mediu de pachete prin fiecare coadă de așteptare. Calculul presupune rezolvarea sistemului de ecuații liniare rezultat pe baza relațiilor de tipul (5.2.14) și descris sub următoarea formă matricială:

$$\mathbf{A} \times \mathbf{\Lambda} = \mathbf{B} \quad (5.2.18)$$

În această ecuație  $A$  este matricea coeficienților, o matrice bidimensională de dimensiune  $N \times N$  scrisă pe baza grafului rețelei și compusă din elementele

$$A_{ij} = c_{ij} \times d_{ij}$$

unde

$$c_{ij} = \begin{cases} 1 & \text{dacă există muchie între } v_i \text{ și } v_j, \text{ respectiv } i = j \\ 0 & \text{în caz contrar} \end{cases}$$

$$d_{ij} = \begin{cases} \frac{1 - \alpha}{\text{nr. muchiilor de ieșire din nodul } v_i} & \text{dacă } i \neq j \\ 1 & \text{dacă } i = j \end{cases}$$

Deoarece am admis că un nod transmite pachete în mod uniform către celelalte  $N-1$  noduri, se poate considera că  $\alpha = 1/(N-1)$ . Matricea unidimensională  $\Lambda$  de  $N \times 1$  este matricea necunoscutelor cu elementele  $\Lambda_i$ . Matricea termenilor liberi  $B$  este o matrice coloană  $N \times 1$ , elementele sale fiind date de:

$$B_i = \frac{\lambda_i}{\text{nr. muchiilor de ieșire din nodul } v_i}$$

Rezolvând sistemul (5.2.18) vor rezulta valorile ratelor medii de emiterie a pachetelor pentru fiecare nod pe muchiile sale incidente, adică  $(A_1, A_2, \dots, A_N)$ . În cazul unei rețele complet conectate (cu legătură între orice pereche de noduri) și dacă  $\lambda_1 = \lambda_2 = \dots = \lambda_N = \lambda$  se obține  $A_1 = A_2 = \dots = A_N = \lambda$ . Condiția de consistență cere ca  $\sum_{v_k \in \mathcal{U}_i} A_k < \mu_i$

( $i = 1, 2, \dots, N$ ), unde  $\mathcal{U}_i$  este mulțimea nodurilor vecine cu  $v_i$ . Pentru cazul particular considerat anterior este necesar ca  $(N-1)\lambda < \mu$ . Cu ajutorul acestor rate se poate calcula pe baza formulelor demonstrate pentru cozile de așteptare oricare din indicatorii aferenți acestor cozi. Deoarece cozile de așteptare sunt asociate legăturilor din rețea, acești indicatori vor reflecta gradul de încărcare cu trafic a fiecărei linii de comunicație, depinzând atât de topologia rețelei, cât și de parametrii dinamici ai fiecărui nod (rata de emiterie a pachetelor și rata de prelucrare a pachetelor primite). Astfel se poate calcula pentru linia de legătură de la nodul  $v_i$  la nodul  $v_j$  probabilitatea ca în coada de așteptare corespunzătoare  $ij$  să nu fie în așteptare nici un pachet, luând în considerare situațiile când sistemul de cozi aferent lui  $v_i$  este gol (relația (5.2.11)), respectiv când acesta conține  $m$  pachete (relația (5.2.16)):

$$p_{ij}^0 = p_j^0 + \sum_{m=1}^{\infty} p_{ij}(m, 0) = (1 - \rho_j^2) + \sum_{m=1}^{\infty} (1 - \rho_j) \rho_j^{m+1} (1 - \gamma_{ij})^m =$$

$$= 1 - \rho_j^2 + \frac{(1 - \rho_j) \rho_j^2 (1 - \gamma_{ij})}{1 - \rho_j (1 - \gamma_{ij})} = \frac{(1 - \rho_j) (1 + \rho_j \gamma_{ij})}{1 - \rho_j (1 - \gamma_{ij})} \quad (5.2.19)$$



În această relație am folosit notațiile  $\rho_j = \frac{\sum_{v_k \in \mathcal{U}_j} A_k}{\mu_j}$  și  $\gamma_{ij} = \frac{A_i}{\sum_{v_k \in \mathcal{U}_j} A_k}$ ,  $\mathcal{U}_j$  fiind mulțimea

nodurilor vecine cu  $v_j$ . În cazul unei rețele complet conectate cu  $N$  noduri și rate identice de emiterie  $\lambda$ , dacă notez  $\delta_j = \frac{\lambda}{\mu_j}$ , relația (5.2.19) ia o formă mai simplă:

$$p_{ij}^0 = (1 + \delta_j) \frac{1 - (N-1)\delta_j}{1 - (N-2)\delta_j} \quad (5.2.20)$$

Mărimea  $p_{ij}^0$  reprezintă de fapt probabilitatea staționară ca linia de la  $v_i$  la  $v_j$  să fie liberă, ea constituind astfel un indicator de performanță al rețelei. Dacă se admite că această legătură este expusă la defectare cu probabilitatea  $q_{ij} = 1 - p_{ij}$ , atunci se poate calcula un indicator reunit de performanță și fiabilitate aferent legăturii, și anume probabilitatea ca aceasta să fie în stare de funcționare și în același timp liberă pentru un transfer:

$$P_{ij} = \Pr\{\text{legătura operațională}\} \cdot \Pr\{\text{nici un pachet în așteptare}\} = p_{ij} p_{ij}^0 \quad (5.2.21)$$

La nivelul rețelei, în loc de indicatorul clasic de fiabilitate de tip terminal care exprimă probabilitatea ca să existe o cale de comunicație între două noduri, voi introduce un indicator de performabilitate, care ține cont și de încărcarea cu trafic a rețelei. Pentru o pereche de noduri  $(s, t)$  definesc *accesibilitatea instantanee* (instantaneous reachability) ca fiind probabilitatea de a exista cel puțin o cale de legătură funcțională de la nodul  $s$  la  $t$  astfel încât de-a lungul acesteia să nu fie nici un pachet în așteptare, adică:

$$I_{st} = \Pr\{\text{există cel puțin o cale operațională liberă de la } s \text{ la } t\} \quad (5.2.22)$$

Desigur că și accesibilitatea instantanee se poate extinde, similar cu fiabilitatea de tip multiterminal, la cazul comunicării simultane între mai multe perechi de noduri. Determinarea lui  $I_{st}$  se poate face cu aceleași metode de calcul ca și fiabilitatea  $R_{st}$  dintr-o rețea, și care au fost amplu prezentate în capitolul 3. Se pornește tot de la funcția de structură, dar în acest caz unei muchii  $x_{ij}$  i se asociază o probabilitate dată de relația (5.2.21), astfel încât  $\Pr\{x_{ij} = 1\} = P_{ij}$ . Pentru a menține independența evenimentelor, am admis ipoteza simplificatoare conform căreia încărcarea cu trafic a unei legături nu influențează încărcarea cu trafic a nici unei alte linii. Este de menționat faptul că la scrierea funcției de structură (adică de cale) trebuie ținut cont de sensul de transmitere, fiindcă unei muchii bidirecționale i se asociază două probabilități și în general  $P_{ij} \neq P_{ji}$ , deoarece încărcarea cu trafic în cele două sensuri este de obicei diferită.

Indicatorul de performabilitate introdus are avantajul că pe lângă caracteristicile de fiabilitate ale rețelei surprinde și dinamica de funcționare a acesteia. Astfel devine posibilă compararea diferitelor soluții sub aspectul reunit al alocării fiabilității componentelor și al încărcării rețelei. Un alt avantaj este faptul că după calculul intensității de trafic și a probabilității asociate fiecărei linii prin procedura prezentată în acest paragraf, determinarea accesibilității instantanee se face cu metodele uzuale de calcul a fiabilității dintr-o rețea.

### 5.3. Validarea modelului

Scopul urmărit a fost întocmirea unui program de simulare a traficului de pachete într-o rețea cu  $n$  noduri, urmând ca datele statistice rezultate din simulare să fie comparate cu cele ale modelului teoretic pentru validarea acestuia. Rețeaua simulată respectă primele trei presupuneri din paragraful precedent, ratele de emisie și de servire a pachetelor fiind constante. În literatura de specialitate există mai multe tehnici de simulare a sistemelor de așteptare [LEE76][VĂDU77], dar particularitatea acestei rețele complexe cu  $n$  servere, fiecare având până la  $n-1$  cozi a necesitat elaborarea unui algoritm propriu, pe care îl prezint pe scurt în continuare. La nivelul unui nod  $i$  ( $i=1, \dots, n$ ) pot apare două feluri de evenimente: emițerea unui pachet (la momentele  $T_{i1}$ ) și terminarea servirii unui pachet (la momentele  $T_{i2}$ ). Ideea de bază constă în determinarea evenimentului următor și avansarea ceasului absolut  $T_{cik}$ , care indică momentul curent, la evenimentul următor, până când se atinge timpul impus pentru simulare  $T_{lim}$ . O descriere formală a algoritmului este următoarea:

simulare\_rețea::

**BEGIN**

*Citește numărul de noduri ( $n$ ) și matricea de conexiuni ;*  
*Citește variabilele de intrare  $\lambda_i, \mu_i$  ( $i=1, \dots, n$ ) și timpul limită  $T_{lim}$  ;*  
*Pregătește generatorul de numere aleatoare ;*  
*Introduce câte un pachet în fiecare server ;*  
*Anulează numărul de pachete din fiecare coadă ;*  
*Inițializează ceasul curent  $T_{cik}=0$  ;*  
*Generează timpii inițiali  $t_{ij}$  ;*  
*Calculează timpii totali  $T_{ij}:=T_{cik}+t_{ij}$*

**REPEAT**

*Determină următorul eveniment  $T_{ik}:=\min(T_{ij}|T_{ij}>T_{cik})$  ;*  
*Actualizează numărul mediu de pachete din fiecare coadă ;*  
*Avansează ceasul  $T_{cik}:=T_{ik}$*

**IF  $l=1$  THEN**

*\* emitere pachet de către nodul  $k$  \*/*  
*Selectează aleator o destinație  $d \neq k$  pentru noul pachet ;*  
*Selectează aleator o linie de ieșire, fie  $m$  ;*  
*Introduce noul pachet în coada  $km$*   
**IF Serverul  $m$  liber THEN**  
*Mută pachetul din coada  $km$  în serverul  $m$  ;*  
*Generează noul  $t_{m2}$  din  $\mu_m$  ;*  
*Calculează noul  $T_{m2}:=T_{cik}+t_{m2}$*

**FI**

*Generează noul  $t_{k1}$  din  $\lambda_k$  ;*  
*Calculează noul  $T_{k1}:=T_{cik}+t_{k1}$*

**FI**

**IF  $l=2$  THEN**

*\* scoatere pachet din serverul  $k$  \*/*  
**IF Pachetul destinat nodului  $k$  THEN**  
*Scoate pachetul din serverul  $k$*

```

ELSE
    Selectează aleator o linie de ieșire, fie  $m$  ;
    Mută pachetul din severul  $k$  în coada  $km$ 
FI
    Selectează următoarea linie de intrare care are pachet, fie  $s$ 
IF Coada  $sk$  nu este goală THEN
    Mută pachetul din coada  $sk$  în serverul  $k$  ;
    Generează noul  $t_{k2}$  din  $\mu_k$  ;
    Calculează noul  $T_{k2} = T_{clk} + t_{k2}$ 
FI FI
UNTIL  $T_{clk} > T_{lim}$  ;
    Afișează rezultatele
END

```

Legat de acest algoritm, care a fost implementat în limbajul Turbo Pascal 7.0, se mai impun câteva precizări:

1. Timpul  $t_{k1}$  între două emiteri de pachete de către nodul  $k$  este conform ipotezei o variabilă aleatoare cu distribuție exponențială de parametru  $\lambda_k$ , care se obține generând un număr aleator uniform  $u$  între 0 și 1 pe baza căruia se calculează timpul cu formula:

$$t_{k1} = -\frac{1}{\lambda_k} \ln u \quad (5.3.1)$$

2. Durata  $t_{k2}$  a unei serviri, o variabilă aleatoare presupusă cu distribuție exponențială de parametru  $\mu_k$ , se calculează în mod similar, cu relația:

$$t_{k2} = -\frac{1}{\mu_k} \ln u \quad (5.3.2)$$

3. Calculul mediei (în acest caz a numărului de pachete din fiecare coadă  $\bar{n}_j$ ) trebuie raportat nu la numărul de iterații, ci ea trebuie actualizată în raport cu momentul evaluării, conform relației:

$$\bar{n}_j = \frac{T_{clk} \bar{n}_j^{vechi} + (T_{ei} - T_{clk}) n_j}{T_{ei}} \quad (5.3.3)$$

4. Pentru selecția cozii de intrare în nodul  $k$  din care se va scoate următorul pachet am adoptat tehnica de priorități rotitoare, ultima coadă servită fiind plasată la urma șirului de prioritate (LSLQ, last-served last-queued). Într-o altă variantă a programului am implementat tehnica preluării pachetelor pentru servire în ordinea sosirilor (FIFO, first-in first-out), asociind fiecărui pachet și un indice care marchează momentul intrării sale în coadă.

În urma experimentelor efectuate cu ajutorul programului de simulare și a comparării rezultatelor obținute cu cele deduse teoretic s-au desprins o mulțime de concluzii interesante. Indicatorul urmărit a fost numărul mediu de pachete aflate în fiecare

coadă de așteptare a rețelei. Rezultatele obținute prin simulare au confirmat aplicabilitatea modelului teoretic la rețelele complet conectate în care nodurile au aceeași rată de emisie a pachetelor. Acest lucru se poate constata și pe baza tabelului 5.3.1, care sintetizează rezultatele pentru rețelele cu 3 și 4 noduri, ratele  $\lambda$  și  $\mu$  fiind alese astfel încât să satisfacă condiția de consistență exprimată în paragraful precedent. Pentru aceste valori am observat o tendință de stabilizare a rezultatelor de simulare la  $T_{\text{sim}} > 5000$  unități de timp. Legat de precizia modelului, două aspecte s-au dovedit a fi relevante: valoarea lui  $\alpha$  și modul de calcul al numărului estimat de pachete conform (5.2.15). În ceea ce privește procentul de pachete destinate unui nod din totalul celor care ajung la el, simularea a arătat că pentru o rețea oarecare  $\alpha$  depinde de poziția topologică a nodului considerat și de ratele de emisie, dar nu este influențată de ratele de prelucrare ale nodurilor. La o rețea complet conectată și cu același  $\lambda$  la fiecare nod, valoarea presupusă  $\alpha = 1/(N-1)$  este exactă, pe când scăderea lui  $\lambda$ , al nodului respectiv duce la creșterea lui  $\alpha$ . Astfel modelul cu același  $\alpha$  pentru fiecare nod rămâne un model aproximativ în aceste rețele, aplicabil doar dacă  $\lambda$ , au valori apropiate. Legat de calculul numărului mediu de pachete în fiecare coadă, simularea a dovedit că relația (5.2.15) este valabilă dacă serverul asociat unui nod prelucrează pachetele din cozile aferente după strategia FIFO. Studiind cu un program auxiliar influența disciplinei de servire asupra numărului mediu de pachete dintr-un sistem cu o singură stație și mai multe cozi, s-a pus în evidență faptul că dacă se aplică tehnica priorităților rotitoare, cozile cu rata mai mare de depunere sunt mai lungi decât valoarea calculată cu relația amintită. Scoaterea pe rând din fiecare coadă face ca extragerea mai frecventă din cozile cu rata mai mică să reducă lungimile acestora. Totuși, dacă ratele  $\lambda$ , ale cozilor sunt identice și numărul mediu de pachete este supraunitar, rezultatele statistice obținute prin simulare nu evidențiază diferențe semnificative.

Model-Simulare	$E[n_{2,12}]$	$E[n_{2,13}]$	$E[n_{2,21}]$	$E[n_{2,23}]$	$E[n_{2,31}]$	$E[n_{2,32}]$
$N=3, \lambda=30$ $\mu_1=61, \mu_2=64,$ $\mu_3=70$	7,03 / 7,05	2,57 / 2,50	29,50 / 29,66	2,57 / 2,52	29,50 / 29,74	7,03 / 7,07
$N=4$ $\lambda=18$ $\mu_1=55$ $\mu_2=56$ $\mu_3=58$ $\mu_4=60$	$E[n_{2,12}]$ 8,67 / 8,60	$E[n_{2,13}]$ 4,18 / 4,12	$E[n_{2,14}]$ 2,70 / 2,79	$E[n_{2,21}]$ 17,67 / 17,72	$E[n_{2,23}]$ 4,18 / 4,10	$E[n_{2,24}]$ 2,70 / 2,72
	$E[n_{2,31}]$ 17,67 / 17,75	$E[n_{2,32}]$ 8,67 / 8,62	$E[n_{2,34}]$ 2,70 / 2,64	$E[n_{2,41}]$ 17,67 / 17,69	$E[n_{2,42}]$ 8,67 / 8,70	$E[n_{2,43}]$ 4,18 / 4,19

$T_{\text{sim}}=10000$

Tabelul 5.3.1.

În continuare voi ilustra modul de evaluare a accesibilității instantanee între nodurile  $v_1$  și  $v_3$  ale rețelei cu  $N=3$  noduri din figura 5.1.3. Funcția de structură corespunzătoare comunicației dintre aceste noduri este:

$$\Phi_{13} = x_{13} \cup x_{12} x_{23} \quad (5.3.4)$$

Aplicând metoda sumei de produse disjuncte (capitolul 3) și relația (5.2.21), expresia indicatorului de accesibilitate instantanee în cazul în care se consideră că toate legăturile au aceeași fiabilitate  $p$ , se poate scrie:

$$I_{13} = P_{13} + P_{12}P_{23}(1 - P_{13}) = pP_{13}^0 + p^2P_{12}^0P_{23}^0 - p^3P_{12}^0P_{23}^0P_{13}^0 \quad (5.3.5)$$

Mai departe, dacă nodurile emit pachetele cu aceeași rată ( $\lambda_1 = \lambda_2 = \lambda_3 = \lambda$ ) și având în vedere (5.2.20), relația precedentă devine:

$$I_{13} = p(1 + \delta_3) \frac{1 - 2\delta_3}{1 + 2\delta_3} \left[ 1 + p(1 + \delta_2) \frac{1 - 2\delta_2}{1 + 2\delta_2} \left( 1 - p(1 + \delta_3) \frac{1 - 2\delta_3}{1 + 2\delta_3} \right) \right] \quad (5.3.6)$$

Dacă se ia  $p=0,995$ ,  $\lambda=8$  și  $\mu_2=\mu_3=20$ , atunci se obține  $I_{13}=0,175$ . Dublarea ratei de prelucrare a calculatorului din nodul  $v_2$  ( $\mu_2=40$ ) duce la creșterea indicatorului la valoarea  $I_{13}=0,221$ , ceea ce denotă așa cum era de așteptat un indice superior al comunicării de la nodul  $v_1$  la  $v_3$ . La acest lucru poate conduce și o diminuare a ratei de emisie, de exemplu pentru  $\lambda=6$  rezultă  $I_{13}=0,394$ .

În ultimele două paragrafe am materializat o idee referitoare la definirea și evaluarea unui indicator de performabilitate care să țină cont de dinamica traficului de pachete într-o rețea modelată printr-un graf neorientat în care nodurilor li se asociază rate de emisie și de prelucrare a pachetelor. Modelul pentru calculul încărcării legăturilor de comunicație a fost validat prin simulare pe calculator, o dezvoltare în continuare a modelului fiind indicată în sensul ca dirijarea pachetelor să țină seama de legăturile defecte. Extinderea modelului și pentru alte genuri de rețele decât cele complet conectate cu rate identice de emisie, prin determinarea exactă a coeficientului  $\alpha$  în fiecare nod pentru aceste cazuri, va putea constitui un obiectiv pentru cercetări viitoare.

## 6. CONCLUZII

Teza de față se încadrează în domeniul studiului comportării sistemelor cu procesoare multiple sub aspectul fiabilității și performanței, justificat de răspândirea largă a acestor sisteme și tratarea lor preferențială în literatura de specialitate a ultimului deceniu. Maniera de structurare adoptată pentru lucrare a permis includerea contribuțiilor personale ale autorului la domeniul menționat, o atenție deosebită fiind acordată verificării pe exemple și validării prin simulare a soluțiilor propuse. Obiectivele principale urmărite și realizate pe parcursul tezei pot fi grupate astfel:

- Clasificarea și delimitarea arhitecturilor de calculatoare distribuite care au făcut obiectul cercetărilor de fiabilitate.
- Asigurarea bazei teoretice a modelării arhitecturilor multiprocesor în scopul evaluării fiabilității acestora.
- Evidențierea tendințelor actuale în fiabilitatea sistemelor cu componente multiple în general și a sistemelor cu procesoare multiple în particular, cu includerea contribuțiilor proprii ale autorului.
- Elaborarea unui model unitar de performanță și fiabilitate pentru tipurile de bază de sisteme multiprocesor, respectiv pentru rețele de calculatoare.
- Validarea cu ajutorul programelor de simulare întocmite de autor a modelelor matematice de performabilitate propuse.

Pomind de la aceste obiective propuse, în cele ce urmează se scot în evidență principalele contribuții originale ale autorului.

1. În primul capitol de consistență care urmează celui introductiv, se urmărește delimitarea arhitecturilor de sisteme cu mai multe procesoare care vor face obiectul analizei și modelării fiabiliste. În acest scop autorul și-a adus contribuția la:

- 1.1. Sistematizarea taxonomiei sistemelor de calcul distribuite pe baza criteriilor de granularitate a interacțiunilor și tehnicilor de transfer a datelor.
- 1.2. Evidențierea caracteristicilor de trafic și toleranță la defecte pentru arhitecturile reprezentative de sisteme cu mai multe procesoare.

2. Paragrafele 3.1-3.5 și 3.7-3.8 sunt dedicate pentru sintetizarea conceptelor fundamentale de fiabilitate aplicabile la sistemele cu mai multe procesoare. În scopul utilității la sistemele strâns cuplate, problema de fiabilitate a fost transpusă din spațiul timpului continuu în cel discret, fiind prezentat și un studiu comparativ asupra distribuțiilor fără memorarea trecutului în cele două cazuri. Este de remarcă utilizarea unitară a funcției de structură scrisă sub formă booleană, distinctă față de cea algebrică, pentru calculul fiabilității sistemelor de tip multiterminal prin metodele analitice de bază. Contribuțiile originale la această parte de fundamente matematice pentru analiza fiabilității sistemelor multiprocesor sunt:

- 2.1. Expriamarea indicatorilor de fiabilitate în spațiul timpului discret.
- 2.2. Tratarea matematică originală prin teorema 3.3.2 a cazului când funcția de fiabilitate de variabilă discretă are forma polinomială.

- 2.3. Efectuarea unui studiu comparativ finalizat prin tabelul 3.7.4, din care au fost desprinse concluziile privind complexitatea calculului la metodele analitice de bază de calcul a fiabilității unei rețele.
3. Problematica evaluării fiabilității sistemelor de calcul modelate printr-un graf neorientat de tip serie-paralel prin tehnici de reducere cu menținerea fiabilității a fost completat de autor în paragraful 3.6 cu următoarele:
- 3.1. Adaptarea tehnicilor de reducere și la situația grafurilor orientate, rezultatele fiind sintetizate în tabelul 3.6.3.
  - 3.2. Studiul (finalizat prin tabelul 3.6.6) al cazurilor de descompunere a grafului rețelei când acesta este ireductibil prin transformări simple.
4. Urmărind transpunerea metodelor de analiză și sinteză a circuitelor logice pentru determinarea fiabilității, o parte consistentă a tezei (paragrafele 3.9 și 3.10) se ocupă de utilizarea tehnicii booleene în fiabilitatea rețelelor. Contribuțiile esențiale aduse de autor la această problemă sunt:
- 4.1. Efectuarea unei paralele între procedurile de calcul probabilistic a fiabilității unei rețele și cele bazate pe tehnica minimizării funcțiilor logice pe diagrame Karnaugh.
  - 4.2. Statuarea teoremelor 3.9.1 și 3.9.2 care fac legătura între funcția de structură booleană și setul de căi/tăieturi minimale.
  - 4.3. Elaborarea algoritmului prezentat în anexa 2 pentru minimizarea disjunctă a funcției de structură, care produce expresia de fiabilitate cu numărul cel mai mic de termeni.
  - 4.4. Considerarea căilor de bază și alternative pentru simplificarea determinării expresiei fiabilității de tip terminal dintr-o rețea.
  - 4.5. Elaborarea unei metode sistematice de eliminare a termenilor redundanți la exprimarea funcției de fiabilitate pe bază de sume disjuncte, prezentat pe exemplul 3.10.7.
  - 4.6. Folosirea decompoziției funcției de structură booleene pentru simplificarea calculului expresiei de fiabilitate, ilustrat în exemplul 3.10.12.
5. Lărgirea domeniului de aplicabilitate a relațiilor logice de semnal ca și o metodă alternativă de evaluare a fiabilității unei rețele a făcut obiectul paragrafului 3.11. Elementele originale prezentate de autor în legătură cu acest subiect sunt:
- 5.1. Corectarea unei formule de calcul din literatură [AGGA78, tabelul 1, relația 7] pentru calculul fiabilității pe baza relațiilor logice de semnal, în care autorii nu au acoperit toate stările posibile ale subrețelei considerate.
  - 5.2. Completarea algoritmului de calcul a fiabilității pe baza relațiilor logice de semnal propus în [AGGA78] cu încă un pas în vederea obținerii unei expresii mai simple a funcției de fiabilitate, verificat pe exemplul 3.11.1.
  - 5.3. Stabilirea modalităților de utilizare a relațiilor logice de semnal în cazul unei rețele modelate printr-un graf orientat, rezultatele fiind sintetizate în tabelul 3.11.4.
  - 5.4. Extinderea metodei relațiilor logice de semnal pentru determinarea fiabilității globale într-o rețea orientată, formulele de calcul fiind date în tabelul 3.11.5, iar un model de aplicare în exemplul 3.11.2.
6. Importanța legăturilor dintre procesoare în fiabilitatea rețelei fiind o problemă esențială în procesul de sinteză, autorul a propus și ilustrat prin exemplul 3.12.3 un

procedeu de alocare a fiabilităților pentru compensarea inegalităților de cauză topologică dintr-o rețea.

7. În vederea unei dezvoltări viitoare a tezei prin renunțarea la ipoteza independenței dintre defecte și a comportării bivalente, în paragrafele 3.13 și 3.14 autorul a efectuat sintetizarea tendințelor ce se manifestă în domeniul evaluării fiabilității sistemelor cu defecțiuni dependente, respectiv cu stări multiple.

8. În cadrul capitolului 4 scopul urmărit este studiul sistemelor multiprocesor strâns cuplate sub aspectul reunit de fiabilitate și performanță, admitând că elementele componente ale rețelei de interconectare a sistemului sunt expuse la defectare. În acest context, autorul și-a adus următoarele contribuții originale:

8.1. Stabilirea condiției limită de valabilitate a relațiilor [WILK91, p.260] pentru calculul lățimii de bandă a sistemelor cu magistrale multiple și tratarea exhaustivă a ambelor cazuri posibile în funcție de raportul dintre numărul de procesoare și cel de memorii comune.

8.2. Construirea (în paragraful 4.2) a unui model matematic în timp discret în scopul evaluării performanțelor cu luarea în considerare a aspectelor de fiabilitate și particularizarea sa pentru arhitecturile de bază de tip crossbar, cu magistrală unică, cu magistrale multiple și cu rețea multinivel.

8.3. Definirea în trei feluri distincte și calculul duratei de viață utile a unui sistem crossbar, urmate de interpretarea rezultatelor pe baza tabelului comparativ 4.2.8.

8.4. Conceperea algoritmilor (prezentați în Anexa 4) și elaborarea programelor de simulare a comportării sistemelor multiprocesor de tip crossbar, cu magistrală unică, cu magistrale multiple și cu rețea de interconectare multinivel, bazate pe generarea de evenimente aleatoare.

8.5. Validarea modelului de performabilitate propus prin verificarea întreprinsă în paragraful de rezultate experimentale asupra principalilor indicatori de performanță teoretici și obținuți prin simulare, pe baza a opt grafice comparative.

8.6. Identificarea prin modelul propus a limitei inferioare de prag (evidențiată grafic) pentru fiabilitatea comutatoarelor dintr-o rețea crossbar.

8.7. Evidențierea prin două grafice și analiza numărului de procesoare aflate în așteptare pentru un sistem crossbar, respectiv cu magistrală unică.

8.8. Evidențierea prin două grafice și analiza timpului mediu de viață al sistemelor cu magistrale pentru diferite configurații.

8.9. Evidențierea grafică și analiza probabilității de acceptare a unei cereri în cazul unui sistem cu rețea multinivel de tip baseline.

9. În scopul de a oferi o soluție avantajoasă sub aspectul costului în procesul de sinteză al sistemelor multiprocesor strâns cuplate și utilizând de modelul propus, în cadrul paragrafului 4.3 au fost introduse următoarele elemente originale:

9.1. Definirea unui indicator de eficiență al sistemului crossbar evidențiat grafic pentru trei cazuri particulare, care permite alegerea numărului optim de module de memorie comună.

9.2. Indicarea pe bază grafică a unei metode de determinare a numărului de magistrale necesare într-un sistem multiprocesor cu o rată dată a cererilor de acces.

9.3. Formularea recomandărilor privind alegerea unei anumite arhitecturi multiprocesor pe baza graficului timpului de răspuns minim.



10. Ultimul capitol de consistență al tezei este dedicat analizei sistemelor distribuite slab cuplate într-un cadru unificat de fiabilitate și performanță. Pentru aceasta, după o sinteză a principalelor concepte de evaluare a fiabilității și performanțelor acestor sisteme, autorul urmărește introducerea unui nou indicator de performabilitate, a cărui evaluare este posibilă cu metodele de calcul a fiabilității rețelelor, prezentate în cadrul capitolului 3. În acest context, contribuțiile originale aduse pe parcursul acestui capitol sunt:

10.1. Formularea a trei criterii de alocare a copiilor multiple de fișiere pentru eficientizarea fiabilității într-o rețea de calculatoare.

10.2. Construirea modelului de performabilitate prezentat în paragraful 5.2 pentru o rețea de calculatoare complet conectată, care ia în considerare încărcarea cu trafic a legăturilor de comunicație.

10.3. Introducerea indicatorului de accesibilitate instantanee pentru caracterizarea reunită a performanțelor și fiabilității transmisiei dintre terminalele unei rețele.

10.4. Conceperea algoritmului (dat în paragraful 5.3) și elaborarea programului de simulare pentru validarea modelului teoretic de transmitere a pachetelor de date într-o rețea de calculatoare.

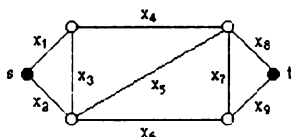
În concluzie se poate afirma că în cadrul problematicii abordate au fost propuse elemente și soluții noi, menite să completeze tratarea matematică și modelarea sub aspect fiabilistic a sistemelor investigate. Prin conținutul său, teza se dorește a fi și un fundament pentru cercetări viitoare, care trebuie să dezvolte aspecte cum ar fi studiul disponibilității sistemelor cu un spațiu foarte larg al stărilor utilizând lanțuri Markov [SAHN87][SHAR93], modelarea efectului dependenței statistice dintre defectele componentelor asupra sistemelor cu stări multiple [VEER93][VEER94] sau găsirea de tehnici pentru includerea impactului factorilor care depind de protocolul utilizat pentru transferul de date [CATA93][COHEN94][SIMM94].

Teza de doctorat are o extindere de 223 pagini, materialul scris fiind ilustrat cu 103 figuri și 16 tabele, iar bibliografia cuprinde 156 de titluri (dintre care 15 ale autorului).

## ANEXA 1

**Determinarea termenilor disjuncti pentru calculul  
fiabilității pe baza vectorilor de cale**

Rețeaua ARPA redusă ( $n = 9$  muchii,  $m = 13$  căi):



$$\Phi = x_1x_4x_8 \cup x_1x_4x_7x_9 \cup x_1x_4x_5x_6x_9 \cup x_1x_3x_5x_8 \cup x_1x_3x_5x_7x_9 \cup x_1x_3x_6x_7x_8 \cup \\ \cup x_1x_3x_6x_9 \cup x_2x_3x_4x_8 \cup x_2x_3x_4x_7x_9 \cup x_2x_5x_8 \cup x_2x_5x_7x_9 \cup x_2x_6x_7x_8 \cup x_2x_6x_9$$

Setul vectorilor de cale minimali (ordonați după cardinalitate):

$P$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$
1	1	-	-	1	-	-	-	1	-
2	-	1	-	-	1	-	-	1	-
3	-	1	-	-	-	1	-	-	1
4	1	-	-	1	-	-	1	-	1
5	1	-	1	-	1	-	-	1	-
6	1	-	1	-	-	1	-	-	1
7	-	1	1	1	-	-	-	1	-
8	-	1	-	-	1	-	1	-	1
9	-	1	-	-	-	1	1	1	-
10	1	-	-	1	1	1	-	-	1
11	1	-	1	-	1	-	1	-	1
12	1	-	1	-	-	1	1	1	-
13	-	1	1	1	-	-	1	-	1

Tabelul de comparații:

Se compară		$X$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$
$P_1$	-	-	1	-	-	1	-	-	-	1	-
$P_2$	$P_1$	$(x_1, x_4)$	0	1	-	-	1	-	-	1	-
			1	1	-	0	1	-	-	1	-
$P_3$	$P_1$	$(x_1, x_4, x_8)$	0	1	-	-	-	1	-	-	1
			1	1	-	0	-	1	-	-	1
			1	1	-	1	-	1	-	0	1
	$P_2$	$(x_5, x_8)$	0	1	-	-	-	1	-	-	1
			0	1	-	-	1	1	-	0	1
			1	1	-	0	0	1	-	-	1
		disjuncti	1	1	-	1	-	1	-	0	1

$P_4$	$P_1$	$(x_8)$	1	-	-	1	-	-	1	0	1	√
	$P_2$	disjuncti	1	-	-	1	-	-	1	0	1	√
	$P_3$	$(x_2, x_6)$	1	0	-	1	-	-	1	0	1	√
			1	1	-	1	-	0	1	0	1	√
$P_5$	$P_1$	$(x_4)$	1	-	1	0	1	-	-	1	-	√
	$P_2$	$(x_2)$	1	0	1	0	1	-	-	1	-	√
	$P_3$	disjuncti	1	0	1	0	1	-	-	1	-	√
	$P_4$	disjuncti	1	0	1	0	1	-	-	1	-	√
$P_6$	$P_1$	$(x_4, x_8)$	1	-	1	0	-	1	-	-	1	√
			1	-	1	1	-	1	-	0	1	√
	$P_2$	$(x_2, x_5, x_8)$	1	0	1	0	-	1	-	-	1	√
			1	1	1	0	0	1	-	-	1	√
			1	1	1	0	1	1	-	0	1	√
		disjuncti	1	-	1	1	-	1	-	0	1	√
	$P_3$	disjuncti	1	0	1	0	-	1	-	-	1	√
		$(x_2)$	1	0	1	1	-	1	-	0	1	√
	$P_4$	disjuncti	1	0	1	0	-	1	-	-	1	√
		$(x_7)$	1	0	1	1	-	1	0	0	1	√
	$P_5$	$(x_5, x_8)$	1	0	1	0	0	1	-	-	1	√
			1	0	1	0	1	1	-	0	1	√
		disjuncti	1	0	1	1	-	1	0	0	1	√
$P_7$	$P_1$	$(x_1)$	0	1	1	1	-	-	-	1	-	√
	$P_2$	$(x_3)$	0	1	1	1	0	-	-	1	-	√
	$P_3$	$(x_6, x_9)$	0	1	1	1	0	0	-	1	-	√
			0	1	1	1	0	1	-	1	0	√
	$P_4$	disjuncti	0	1	1	1	0	0	-	1	-	√
		disjuncti	0	1	1	1	0	1	-	1	0	√
	$P_5$	disjuncti	0	1	1	1	0	0	-	1	-	√
		disjuncti	0	1	1	1	0	1	-	1	0	√
$P_6$	disjuncti	0	1	1	1	0	0	-	1	-	√	
	disjuncti	0	1	1	1	0	1	-	1	0	√	
$P_8$	$P_1$	$(x_1, x_4, x_8)$	0	1	-	-	1	-	1	-	1	√
			1	1	-	0	1	-	1	-	1	√
			1	-	1	1	-	1	0	1	√	
	$P_2$	$(x_8)$	0	1	-	-	1	-	1	0	1	√
		$(x_8)$	1	1	-	0	1	-	1	0	1	√
		disjuncti	1	1	-	1	1	-	1	0	1	√
	$P_3$	$(x_6)$	0	1	-	-	1	0	1	0	1	√
		$(x_6)$	1	1	-	0	1	0	1	0	1	√
		$(x_6)$	1	1	-	1	1	0	1	0	1	√
	$P_4$	disjuncti	0	1	-	-	1	0	1	0	1	√
		disjuncti	1	1	-	0	1	0	1	0	1	√
	$P_5$	disjuncti	0	1	-	-	1	0	1	0	1	√
		disjuncti	1	1	-	0	1	0	1	0	1	√
	$P_6$	disjuncti	0	1	-	-	1	0	1	0	1	√
		disjuncti	1	1	-	0	1	0	1	0	1	√
$P_7$	disjuncti	0	1	-	-	1	0	1	0	1	√	
	disjuncti	1	1	-	0	1	0	1	0	1	√	

$P_9$	$P_1$	$(x_1, x_4)$	0 1 - - - 1 1 1 -	√
			1 1 - 0 - 1 1 1 -	√
	$P_2$	$(x_3)$	0 1 - - 0 1 1 1 -	√
		$(x_5)$	1 1 - 0 0 1 1 1 -	√
	$P_3$	$(x_9)$	0 1 - - 0 1 1 1 0	√
		$(x_9)$	1 1 - 0 0 1 1 1 0	√
	$P_4$	disjuncti	0 1 - - 0 1 1 1 0	√
		disjuncti	1 1 - 0 0 1 1 1 0	√
	$P_5$	disjuncti	0 1 - - 0 1 1 1 0	√
		disjuncti	1 1 - 0 0 1 1 1 0	√
$P_6$	disjuncti	0 1 - - 0 1 1 1 0	√	
	disjuncti	1 1 - 0 0 1 1 1 0	√	
$P_7$	$(x_3, x_4)$		0 1 0 - 0 1 1 1 0	√
			0 1 1 0 0 1 1 1 0	√
	disjuncti	1 1 - 0 0 1 1 1 0	√	
$P_8$	disjuncti	0 1 0 - 0 1 1 1 0	√	
	disjuncti	0 1 1 0 0 1 1 1 0	√	
	disjuncti	1 1 - 0 0 1 1 1 0	√	
$P_{10}$	$P_1$	$(x_8)$	1 - - 1 1 1 - 0 1	√
	$P_2$	disjuncti	1 - - 1 1 1 - 0 1	√
	$P_3$	$(x_2)$	1 0 - 1 1 1 - 0 1	√
	$P_4$	$(x_7)$	1 0 - 1 1 1 0 0 1	√
	$P_5$	disjuncti	1 0 - 1 1 1 0 0 1	√
	$P_6$	$(x_3)$	1 0 0 1 1 1 0 0 1	√
	$P_7$	disjuncti	1 0 0 1 1 1 0 0 1	√
	$P_8$	disjuncti	1 0 0 1 1 1 0 0 1	√
	$P_9$	disjuncti	1 0 0 1 1 1 0 0 1	√
$P_{11}$	$P_1$	$(x_4, x_8)$	1 - 1 0 1 - 1 - 1	√
			1 - 1 1 1 - 1 0 1	√
	$P_2$	$(x_2, x_8)$	1 0 1 0 1 - 1 - 1	√
			1 1 1 0 1 - 1 0 1	√
		disjuncti	1 - 1 1 1 - 1 0 1	√
	$P_3$	disjuncti	1 0 1 0 1 - 1 - 1	√
		$(x_6)$	1 1 1 0 1 0 1 0 1	√
		$(x_2, x_6)$	1 0 1 1 1 - 1 0 1	√
			1 1 1 1 1 0 1 0 1	√
	$P_4$	disjuncti	1 0 1 0 1 - 1 - 1	√
		disjuncti	1 1 1 0 1 0 1 0 1	√
	$P_5$	$(x_8)$	1 0 1 0 1 - 1 0 1	√
		disjuncti	1 1 1 0 1 0 1 0 1	√
	$P_6$	$(x_6)$	1 0 1 0 1 0 1 0 1	√
disjuncti		1 1 1 0 1 0 1 0 1	√	
$P_7$	disjuncti	1 0 1 0 1 0 1 0 1	√	
	disjuncti	1 1 1 0 1 0 1 0 1	√	
$P_8$	disjuncti	1 0 1 0 1 0 1 0 1	√	
$P_9$	disjuncti	1 0 1 0 1 0 1 0 1	√	
$P_{10}$	disjuncti	1 0 1 0 1 0 1 0 1	√	

$P_{12}$	$P_1$	$(x_4)$	1	-	1	0	-	1	1	1	-	√
	$P_2$	$(x_2, x_3)$	1	0	1	0	-	1	1	1	-	√
			1	1	1	0	0	1	1	1	-	√
	$P_3$	disjuncti	1	0	1	0	-	1	1	1	-	√
		$(x_9)$	1	1	1	0	0	1	1	1	0	√
	$P_4$	disjuncti	1	0	1	0	-	1	1	1	-	√
		disjuncti	1	1	1	0	0	1	1	1	0	√
	$P_5$	$(x_5)$	1	0	1	0	0	1	1	1	-	√
		disjuncti	1	1	1	0	0	1	1	1	0	√
	$P_6$	$(x_9)$	1	0	1	0	0	1	1	1	0	√
		disjuncti	1	1	1	0	0	1	1	1	0	√
$P_7$	disjuncti	1	0	1	0	0	1	1	1	0	√	
	disjuncti	1	1	1	0	0	1	1	1	0	√	
$P_8$	disjuncti	1	0	1	0	0	1	1	1	0	√	
	disjuncti	1	1	1	0	0	1	1	1	0	√	
$P_9$	disjuncti	1	0	1	0	0	1	1	1	0	√	
$P_{10}$	disjuncti	1	0	1	0	0	1	1	1	0	√	
$P_{11}$	disjuncti	1	0	1	0	0	1	1	1	0	√	
$P_{13}$	$P_1$	$(x_1, x_8)$	0	1	1	1	-	-	1	-	1	√
			1	1	1	1	-	-	1	0	1	√
	$P_2$	$(x_5, x_8)$	0	1	1	1	0	-	1	-	1	√
			0	1	1	1	1	-	1	0	1	√
		disjuncti	1	1	1	1	-	-	1	0	1	√
	$P_3$	$(x_6)$	0	1	1	1	0	0	1	-	1	√
		$(x_6)$	0	1	1	1	1	0	1	0	1	√
		$(x_6)$	1	1	1	1	-	0	1	0	1	√
	$P_4$	disjuncti	0	1	1	1	0	0	1	-	1	√
		disjuncti	0	1	1	1	1	0	1	0	1	√
	$P_5$	disjuncti	0	1	1	1	0	0	1	-	1	√
		disjuncti	0	1	1	1	1	0	1	0	1	√
$P_6$	disjuncti	0	1	1	1	0	0	1	-	1	√	
	disjuncti	0	1	1	1	1	0	1	0	1	√	
$P_7$	$(x_8)$	0	1	1	1	0	0	1	0	1	√	
	disjuncti	0	1	1	1	1	0	1	0	1	√	
$P_8$	disjuncti	0	1	1	1	0	0	1	0	1	√	
$P_9$	disjuncti	0	1	1	1	0	0	1	0	1	√	
$P_{10}$	disjuncti	0	1	1	1	0	0	1	0	1	√	
$P_{11}$	disjuncti	0	1	1	1	0	0	1	0	1	√	
$P_{12}$	disjuncti	0	1	1	1	0	0	1	0	1	√	

Setul complet de termeni disjuncti:

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$
1	-	-	1	-	-	-	1	-
0	1	-	-	1	-	-	1	-
1	1	-	0	1	-	-	1	-
0	1	-	-	0	1	-	-	1
0	1	-	-	1	1	-	0	1
1	1	-	0	0	1	-	-	1

1	1	-	0	1	1	-	0	1
1	1	-	1	-	1	-	0	1
1	0	-	1	-	-	1	0	1
1	1	-	1	-	0	1	0	1
1	0	1	0	1	-	-	1	-
1	0	1	0	0	1	-	-	1
1	0	1	0	1	1	-	0	1
1	0	1	1	-	1	0	0	1
0	1	1	1	0	0	-	1	-
0	1	1	1	0	1	-	1	0
0	1	-	-	1	0	1	0	1
1	1	-	0	1	0	1	0	1
0	1	0	-	0	1	1	1	0
0	1	1	0	0	1	1	1	0
1	1	-	0	0	1	1	1	0
1	0	0	1	1	1	0	0	1
1	0	1	0	1	0	1	0	1
1	0	1	0	0	1	1	1	0
0	1	1	1	0	0	1	0	1

Funcția de fiabilitate (cu notațiile  $\Pr\{x_i=1\} = p, q = 1 - p$ ):

$$R_{\pi} = p^3 + p^3q + p^4q + p^3q^2 + 4p^4q^2 + 2p^5q^2 + p^5q + 3p^4q^3 + 5p^5q^3 + p^4q^4 + 5p^5q^4$$

## ANEXA 2

### Determinarea expresiei de fiabilitate prin minimizarea disjunctă a funcției de structură booleene

Se consideră funcția de structură  $\Phi(x_1, x_2, x_3, x_4, x_5, x_6)$  dată sub forma canonică disjunctivă, scrisă ca și suma indicilor termenilor minimali corespunzători succesului (stărilor operaționale ale sistemului):

$$\Phi = \sum_6 (21, 23, 29, 31, 45, 47, 53, 55, 61, 63)$$

Metoda constă din două etape:

A. Determinarea tuturor implicațiilor, prin ordonarea termenilor pe grupe în funcție de ponderile binare și aplicarea următoarelor reguli pentru stabilirea adiacenței a doi termeni:

- (1) diferența ponderilor este 1
- (2) diferența indicilor este o putere a lui 2
- (3) termenul cu ponderea mai mare dispune și de indicele mai mare

B. Determinarea setului disjunct de implicații primi, de ordinul cât mai mare posibil, bazat pe următoarele reguli:

- (1) lista se parcurge în sensul descrescător al ordinelor și implicații care au un termen inclus într-un implicant de ordin superior se vor elimina
- (2) în cadrul unui ordin se rețin acei implicații care cauzează eliminarea celui mai mic număr de implicații de același ordin
- (3) în caz de egalitate la criteriul (2), alegerea este indiferentă (formele minime posibile vor avea aceeași complexitate)

Schema de minimizare pentru funcția dată (în paranteze s-au trecut diferențele indicilor, care reprezintă ponderile variabilelor eliminate):

Ordinul 0	Ordinul 1																
Grup 3 <u>21</u>	Grup 3 <u>21, 23 (2)</u>																
Grup 4 <u>23</u>	21, 29 (8)																
29	<u>21, 53 (32)</u>																
45	Grup 4 <u>23, 31 (8)</u>																
<u>53</u>	23, 55 (32)																
Grup 5 <u>31</u>	29, 31 (2)																
47	29, 61 (32)																
55	45, 47 (2)																
<u>61</u>	45, 61 (16)																
Grup 6 <u>63</u>	53, 55 (2)																
	<u>53, 61 (8)</u>																
	Grup 5 <u>31, 63 (32)</u>																
	47, 63 (32)																
	55, 63 (8)																
	61, 63 (2)																
<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; border-bottom: 1px solid black;">Ordinul 2</th> <th style="text-align: center; border-bottom: 1px solid black;">Ordinul 3</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Grup 3    <u>21, 23, 29, 31 (2, 8)</u></td> <td style="text-align: center;">Grup 3    <u>21, 23, 29, 31, 53, 55, 61, 63 (2, 8, 32)</u></td> </tr> <tr> <td style="text-align: center;">          21, 23, 53, 55 (2, 32)</td> <td></td> </tr> <tr> <td style="text-align: center;">          <u>21, 29, 53, 61 (8, 32)</u></td> <td></td> </tr> <tr> <td style="text-align: center;">Grup 4    <u>23, 31, 55, 63 (8, 32)</u></td> <td></td> </tr> <tr> <td style="text-align: center;">          29, 31, 61, 63 (2, 32)</td> <td></td> </tr> <tr> <td style="text-align: center;">          45, 47, 61, 63 (2, 16)</td> <td></td> </tr> <tr> <td style="text-align: center;">          <u>53, 55, 61, 63 (2, 8)</u></td> <td></td> </tr> </tbody> </table>		Ordinul 2	Ordinul 3	Grup 3 <u>21, 23, 29, 31 (2, 8)</u>	Grup 3 <u>21, 23, 29, 31, 53, 55, 61, 63 (2, 8, 32)</u>	21, 23, 53, 55 (2, 32)		<u>21, 29, 53, 61 (8, 32)</u>		Grup 4 <u>23, 31, 55, 63 (8, 32)</u>		29, 31, 61, 63 (2, 32)		45, 47, 61, 63 (2, 16)		<u>53, 55, 61, 63 (2, 8)</u>	
Ordinul 2	Ordinul 3																
Grup 3 <u>21, 23, 29, 31 (2, 8)</u>	Grup 3 <u>21, 23, 29, 31, 53, 55, 61, 63 (2, 8, 32)</u>																
21, 23, 53, 55 (2, 32)																	
<u>21, 29, 53, 61 (8, 32)</u>																	
Grup 4 <u>23, 31, 55, 63 (8, 32)</u>																	
29, 31, 61, 63 (2, 32)																	
45, 47, 61, 63 (2, 16)																	
<u>53, 55, 61, 63 (2, 8)</u>																	

Implicații primi rămași dau funcția de structură sub formă de sumă de produse disjuncte:

$$\Phi = x_2 x_4 x_6 \cup x_1 \bar{x}_2 x_3 x_4 x_6$$

Rezultă direct expresia funcției de fiabilitate:

$$R = p_2 p_4 p_6 + p_1 q_2 p_3 p_4 p_6$$

## ANEXA 3

**Ordonarea legăturilor după importanța lor  
în fiabilitatea s-t**

Rețeaua din figura 3.12.1.a are funcția de fiabilitate dată de (3.12.15), pe baza căreia rezultă (pentru fiabilități egale  $p$ ):

$$R(G|x_1=0) = -p^5 + p^4 + p^2$$

$$R(G|x_2=0) = -p^5 + 2p^3$$

$$R(G|x_3=0) = -p^5 + p^3 + p^2$$

$$R(G|x_4=0) = -p^4 + p^3 + p^2$$

$$R(G|x_5=0) = -p^5 + p^4 + p^3$$

$$R(G|x_6=0) = -p^4 + p^3 + p^2$$

$$R(G|x_1=1) = p^5 - 2p^4 - p^3 + 3p^2$$

$$R(G|x_2=1) = p^5 - 3p^4 + 2p^3 + p$$

$$R(G|x_3=1) = p^5 - 3p^4 + p^3 + p^2$$

$$R(G|x_4=1) = 2p^5 - 5p^4 + 2p^3 + 2p^2$$

$$R(G|x_5=1) = p^5 - 2p^4 + p^2 + p$$

$$R(G|x_6=1) = 2p^5 - 5p^4 + 2p^3 + 2p^2$$

Comparând muchiile două câte două și având în vedere că  $0 < p < 1$ , se obține:

- $e_1$  cu  $e_2$

$$R(G|x_1=0) - R(G|x_2=0) = p^2(p^2 + 1 - 2p) = p^2(1 - p)^2 > 0$$

$$R(G|x_2=1) - R(G|x_1=1) = p(-p^3 + 3p^2 - 3p + 1) = p(1 - p)^3 > 0$$

$$\Rightarrow e_1 < e_2$$

- $e_1$  cu  $e_3$

$$R(G|x_3=0) - R(G|x_1=0) = p^3(1 - p) > 0$$

$$R(G|x_1=1) - R(G|x_3=1) = p^2(p^2 - 2p + 2) = p^2[(1 - p)^2 + 1] > 0$$

$$\Rightarrow e_1 > e_3$$

- $e_1$  cu  $e_4$

$$R(G|x_4=0) - R(G|x_1=0) = p^3(p^2 - 2p + 1) = p^3(1 - p)^2 > 0$$

$$R(G|x_1=1) - R(G|x_4=1) = p^2(-p^3 + 3p^2 - 3p + 1) = p^2(1 - p)^3 > 0$$

$$\Rightarrow e_1 > e_4$$

- $e_1$  cu  $e_5$

$$R(G|x_1=0) - R(G|x_5=0) = p^2(1 - p) > 0$$

$$R(G|x_5=1) - R(G|x_1=1) = p(p^2 - 2p + 1) = p(1 - p)^2 > 0$$

$$\Rightarrow e_1 < e_5$$

- $e_1$  cu  $e_6$  (ca la  $e_1$  cu  $e_4$ )

$$R(G|x_6=0) - R(G|x_1=0) = p^3(1 - p)^2 > 0$$

$$R(G|x_1=1) - R(G|x_6=1) = p^2(1 - p)^3 > 0$$

$$\Rightarrow e_1 > e_6$$

- $e_2$  cu  $e_3$

$$R(G|x_3=0) - R(G|x_2=0) = p^2(1 - p) > 0$$

$$R(G|x_2=1) - R(G|x_3=1) = p(p^2 - p + 1) = p[(1 - p)^2 + p] > 0$$

$$\Rightarrow e_2 > e_3$$

- $e_2$  cu  $e_4$

$$R(G|x_4=0) - R(G|x_2=0) = p^2(p^3 - p^2 - p + 1) = p^2[-p^2(1 - p) + (1 - p)] =$$

$$= p^2(1 - p)^2(1 + p) > 0$$

$$R(G|x_2=1) - R(G|x_4=1) = p(-p^4 + 2p^3 - 2p + 1) = p[(1 - p^2)(1 + p^2) -$$

$$- 2p(1 - p^2)] = p(1 - p^2)(1 - p)^2 > 0$$



- $\Rightarrow e_2 > e_4$
- $e_2$  cu  $e_5$ 

$$R(G|x_2=0) - R(G|x_5=0) = p^3(1-p) > 0$$

$$R(G|x_3=1) - R(G|x_2=1) = p^2(p^2 - 2p + 1) = p^2(1-p)^2 > 0$$

$$\Rightarrow e_2 < e_5$$
  - $e_2$  cu  $e_6$  (ca la  $e_2$  cu  $e_4$ )
 
$$R(G|x_6=0) - R(G|x_2=0) = p^2(1-p)^2(1+p) > 0$$

$$R(G|x_2=1) - R(G|x_6=1) = p(1-p^2)(1-p)^2 > 0$$

$$\Rightarrow e_2 > e_6$$
  - $e_3$  cu  $e_4$ 

$$R(G|x_3=0) - R(G|x_4=0) = p^4(1-p) > 0$$

$$R(G|x_4=1) - R(G|x_3=1) = p^2(p^3 - 2p^2 + p + 1) = p^2[p^3 + p(1-p) + (1-p)(1+p)] > 0$$

$$\Rightarrow e_3 < e_4$$
  - $e_3$  cu  $e_5$ 

$$R(G|x_3=0) - R(G|x_5=0) = p^2(1-p^2) > 0$$

$$R(G|x_5=1) - R(G|x_3=1) = p(p^3 - p^2 + 1) = p[-p^2(1-p) + (1-p) + p] =$$

$$= p[(1-p)^2(1+p) + p] > 0$$

$$\Rightarrow e_3 < e_5$$
  - $e_3$  cu  $e_6$  (ca la  $e_3$  cu  $e_4$ )
 
$$R(G|x_3=0) - R(G|x_6=0) = p^4(1-p) > 0$$

$$R(G|x_6=1) - R(G|x_3=1) = p^2[p^3 + (1-p)(1+2p)] > 0$$

$$\Rightarrow e_3 < e_6$$
  - $e_4$  cu  $e_5$ 

$$R(G|x_4=0) - R(G|x_5=0) = p^2(p^3 - 2p^2 + 1) = p^2[-p^2(1-p) - (1-p^2)] =$$

$$= p^2(1-p)(1+p-p^2) = p^2(1-p)[1+p(1-p)] > 0$$

$$R(G|x_5=1) - R(G|x_4=1) = p(-p^4 + 3p^3 - 2p^2 - p + 1) = p[p^3(1-p) -$$

$$- 2p^2(1-p) + (1-p)] = p(1-p)(p^3 - 2p^2 + 1) =$$

$$= p(1-p)[-p^2(1-p) + (1-p^2)] = p(1-p)^2(1+p -$$

$$- p^2) = p(1-p)^2[1+p(1-p)] > 0$$

$$\Rightarrow e_4 < e_5$$
  - $e_4$  cu  $e_6$ 

$$R(G|x_4=0) - R(G|x_6=0) = 0$$

$$R(G|x_4=1) - R(G|x_6=1) = 0$$

$$\Rightarrow e_4 \approx e_6$$
  - $e_5$  cu  $e_6$  (ca la  $e_5$  cu  $e_4$ )
 
$$R(G|x_6=0) - R(G|x_5=0) = p^2(1-p)[1+p(1-p)] > 0$$

$$R(G|x_5=1) - R(G|x_6=1) = p(1-p)^2[1+p(1-p)] > 0$$

$$\Rightarrow e_5 > e_6$$

Rezultatul ordonării este prezentat sub formă de diagramă Hasse în figura 3.12.1.b.

## ANEXA 4

### Algoritmii de simulare a comportării sistemelor multiprocesor

Programele de simulare utilizează o variabilă aleatoare de tip Bernoulli care ia valoarea 1 cu o probabilitate  $z$  și se generează în felul următor:

```
bernoulli(z)::
  BEGIN
    Generează variabila uniformă  $u \in (0,1)$ 
    IF  $u < z$  THEN bernoulli(z):=1
    ELSE bernoulli(z):=0
  END
```

O iterație reprezintă simularea comportării sistemului până la starea din momentul discret de timp  $t$  ( $t = 1, 2, \dots$ ). Indicatorii de performanță urmăriți sunt numărul mediu de memorii ocupate și de procesoare active. Notațiile corespund celor din paragraful 4.2.

```
simulare_crossbar_1(t)::
  /* fără reluarea cererilor neservite */
  BEGIN
    Citește numărul de procesoare (n) și de memorii (m) ;
    Citește rata de acces (α) și rata de fiabilitate (p) ;
    Citește numărul de iterații (iter)
    FOR l:=1 TO iter DO
      Inițializează toate comutatoarele bune
      FOR k:=1 TO t DO
        REPEAT
          IF comutator=bun THEN
            IF bernoulli(p)=0 THEN comutator:=-defect
          FI FI
        UNTIL Toate comutatoarele parcurse
      OD
      Inițializează toate memoriile libere
      REPEAT
        IF bernoulli(α)=1 THEN procesor:=cere
        ELSE procesor:=activ
        FI
        IF procesor=cere THEN
          Selectează uniform o memorie
          IF (memorie=liber) ∧ (comutator=bun) THEN
            procesor:=primește ;
            memorie:=ocupat
          FI FI
        UNTIL Toate procesoarele parcurse ;
      Calculează indicatorii de performanță curenți ;
```

*Actualizează valorile medii ale indicatorilor pentru iterațiile efectuate*  
**OD**  
*Furnizează rezultatele*  
**END**

**simulare\_crossbar\_2(t)::**

*/\* cu reluarea cererii neservite dacă refuzul s-a datorat coliziunii cu o altă cerere \*/*

**BEGIN**

*Citește numărul de procesoare (n) și de memorii (m) ;*

*Citește rata de acces ( $\alpha$ ) și rata de fiabilitate (p) ;*

*Citește numărul de iterații (iter)*

**FOR** l=1 **TO** iter **DO**

*Inițializează toate comutatoarele bune ;*

*Inițializează toate procesoarele active ;*

*Inițializează toate refuzurile din cauza coliziunii*

**FOR** k=1 **TO** t **DO**

**REPEAT**

**IF** comutator=bun **THEN**

**IF** bernoulli(p)=0 **THEN** comutator:=defect

**FI FI**

**UNTIL** Toate comutatoarele parcurse ;

*Inițializează toate memoriile libere*

**REPEAT**

**IF** (procesor≠cere)  $\vee$  ((procesor=cere)  $\wedge$

(cauza\_refuz=com\_defect)) **THEN**

**IF** bernoulli( $\alpha$ )=1 **THEN** procesor:=cere

**ELSE** procesor:=activ

**FI**

*Selectează uniform o memorie ;*

*Asignează memoria la procesor*

**FI**

**IF** procesor=cere **THEN**

**IF** comutator=defect **THEN** cauza\_refuz:=com\_defect

**ELSE**

**IF** memorie = ocupată **THEN** cauza\_refuz:=coliziune

**ELSE**

procesor:=primește ;

memorie:=ocupată

**FI FI FI**

**UNTIL** Toate procesoarele parcurse

**OD**

*Calculează indicatorii de performanță curenți ;*

*Actualizează valorile medii ale indicatorilor pentru iterațiile efectuate*

**OD**

*Furnizează rezultatele*

**END**

**simulare\_magistrală\_unică(t)::**

*/\* cu reluarea cererii unui procesor în așteptare dacă refuzul a fost cauzat de coliziunea cu altă cerere și nu de magistrala defectă \*/*

**BEGIN**

*Citește numărul de procesoare (n) ;*

*Citește rata de acces ( $\alpha$ ) și rata de fiabilitate (p) ;*

*Citește numărul de iterații (iter) ;*

**FOR** l=1 **TO** iter **DO**

*Inițializează magistrala bună ;*

*Inițializează toate procesoarele active ;*

*Inițializează toate refuzurile din cauza coliziunii*

**FOR** k=1 **TO** t **DO**

**IF** magistrala=bună **THEN**

**IF** bernoulli(p)=0 **THEN** magistrala:=defectă

**FI FI**

*Inițializează memoria liberă*

**REPEAT**

**IF** (procesor $\neq$ cere)  $\vee$  ((procesor=cere)  $\wedge$   
(cauza\_refuz=mag\_defectă)) **THEN**

**IF** bernoulli( $\alpha$ )=1 **THEN** procesor:=cere

**ELSE** procesor:=activ

**FI FI**

**IF** procesor=cere **THEN**

**IF** magistrala=defectă **THEN** cauza\_refuz:=mag\_defectă

**ELSE**

**IF** memorie=ocupată **THEN** cauza\_refuz:=coliziune

**ELSE**

*procesor:=primește ;*

*memorie:=ocupată*

**FI FI FI**

**UNTIL** Toate procesoarele parcurse

**OD**

*Calculează indicatorii de performanță curenți ;*

*Actualizează valorile medii ale indicatorilor pentru iterațiile efectuate*

**OD**

*Furnizează rezultatele*

**END**

**simulare\_magistrale\_multiple(t)::**

*/\* cu reluarea cererii la aceeași memorie dacă refuzul a fost cauzat de coliziunea cu altă cerere \*/*

**BEGIN**

*Citește numărul de procesoare (n), de memorii (m) și de magistrale (b) ;*

*Citește rata de acces ( $\alpha$ ) și rata de fiabilitate (p) ;*

*Citește numărul de iterații (iter)*

**FOR** l=1 **TO** iter **DO**

*Inițializează toate magistralele bune ;*

*Inițializează toate procesoarele active ;*

```

Inițializează toate refuzurile din cauza coliziunii
FOR k=1 TO t DO
  REPEAT
    IF magistrala=bună THEN
      IF bernoulli(p)=0 THEN magistrala:=defectă
    FI FI
  UNTIL Toate magistralele parcurse ;
  Inițializează toate memoriile libere ;
  Inițializează toate magistralele libere
  REPEAT
    IF (procesor≠cere) ∨ ((procesor=cere) ∧
      (cauza_refuz=mag_defectă)) THEN
      IF bernoulli(α)=1 THEN procesor:=cere
      ELSE procesor:=activ
      FI
      Selectează uniform o memorie ;
      Asignează memoria la procesor
    FI
    IF procesor=cere THEN
      IF toate magistralele defecte THEN cauza_refuz:=mag_defectă
      ELSE
        IF memorie=ocupată THEN cauza_refuz:=coliziune
        ELSE
          Selectează o magistrală liberă și bună
          IF nu există magistrală liberă THEN cauza_refuz:=coliziune
          ELSE
            procesor:=primește ;
            memorie:=ocupată ;
            magistrală:=ocupată
          FI FI FI FI
        UNTIL Toate procesoarele parcurse
      OD
      Calculează indicatorii de performanță curenți ;
      Actualizează valorile medii ale indicatorilor pentru iterațiile efectuate
    OD
  Furnizează rezultatele
END

```

simulare\_multinivel(t)::

*/\* pentru rețea cu celule de comutare 2×2, cu reluarea cererii neservite la aceeași memorie în caz de coliziune cu altă cerere \*/*

```

BEGIN
  Citește numărul de procesoare (n) și matricile de adiacență (Ai, i=1,...,log2n) ;
  Citește rata de acces (α) și rata de fiabilitate (p) ;
  Citește numărul de iterații (iter)
  FOR l=1 TO iter DO
    Inițializează toate comutatoarele bune ;
    Inițializează toate procesoarele active ;

```

```

Inițializează toate refuzurile din cauza coliziunii
FOR k=1 TO t DO
  REPEAT
    IF comutator=bun THEN
      IF bernoulli(p)=0 THEN comutator:=defect
    FI FI
  UNTIL Toate comutatoarele parcurse ;
  Inițializează toate memoriile libere
  REPEAT
    IF (procesor≠cere) ∨ ((procesor=cere) ∧
      (cauza_refuz=com_defect)) THEN
      IF bernoulli(α)=1 THEN procesor:=cere
      ELSE procesor:=activ
      FI
      Selectează uniform o memorie ;
      Asignează memoria la procesor
    FI
  UNTIL Toate procesoarele parcurse ;
  Inițializează toate intrările și ieșirile libere ;
  Atribuie ieșirilor nivelului 0 starea procesoarelor
  REPEAT
    Transmite stările ieșirilor nivelului precedent la intrările
      nivelului curent conform Ai
  REPEAT
    IF intrare=cere THEN
      Stabilește ieșirea aferentă memoriei destinație
      IF comutator=defect THEN cauza_refuz:=com_defect
      ELSE
        IF ieșire=liberă THEN
          ieșire:=ocupată ;
          Transmite starea intrării la ieșire
        ELSE cauza_refuz:=coliziune
      FI FI FI
    UNTIL Toate intrările parcurse
  UNTIL Toate nivelurile parcurse ;
  Atribuie memoriilor starea ieșirilor ultimului nivel
  REPEAT
    IF memorie=ocupată THEN procesor(asignat memoriei):=primește
    FI
  UNTIL Toate memoriile parcurse
OD
  Calculează indicatorii de performanță curenți ;
  Actualizează valorile medii ale indicatorilor pentru iterațiile efectuate
OD
  Furnizează rezultatele
END

```

## BIBLIOGRAFIE

- [ABRA79] Abraham, J.A., An improved algorithm for network reliability, IEEE Transactions on Reliability, Vol.28, No.1, Apr.1979, p.58-61.
- [AGGA75] Aggarwal, K.K., Gupta, J.S., Misra, K.B., Computational time and absolute error comparison for reliability expression derived by various methods, Microelectronics and Reliability, Vol.14, 1975, p.465-467.
- [AGGA78] Aggarwal, K.K., Rai, S., Symbolic reliability evaluation using logical signal relations, IEEE Transactions on Reliability, Vol.27, No.3, Aug.1978, p.202-205.
- [AGGA81] Aggarwal, K.K., Rai, S., Reliability evaluation in computer communication networks, IEEE Transactions on Reliability, Vol.30, No.1, Apr.1981, p.32-35.
- [AGGA88] Aggarwal, K.K., A fast algorithm for the performance index of a telecommunication network, IEEE Transactions on Reliability, Vol.37, No.1, Apr.1988, p.65-69.
- [AGRA85] Agrawal, D.P., Leu, J.-S., Dynamic accessibility testing and path length optimization of multistage interconnection networks, IEEE Transactions on Computers, Vol.34, No.3, Mar.1985, p.255-266.
- [AJMO86] Ajmone Marsan, M., Balbo, G., Conte, G., Performance Models of Multiprocessor Systems, The MIT Press, Massachusetts, 1986.
- [ANDE89] Anderson, A.J., Multiple Processing, Prentice-Hall, 1989.
- [ARATÓ90] Arató, P., Logikai rendszerek tervezése, Tankönyvkiadó, Budapest, 1990.
- [ARUN79] Arunkumar, S., Lee, S.H., Enumeration of all minimal cut-sets for a node pair in a graph, IEEE Transactions on Reliability, Vol.28, No.1, Apr.1979, p.51-55.
- [ASSO86] Assous, J.Y., First- and second-order bounds on terminal reliability, Networks, Vol.16, 1986, p.319-329.
- [AVEN85] Aven, T., Reliability evaluation of multistate systems with multistate components, IEEE Transactions on Reliability, Vol.34, No.5, Dec.1985, p.473-479.
- [BALL86] Ball, M.O., Computational complexity of network reliability analysis: an overview, IEEE Transactions on Reliability, Vol.35, No.3, Aug.1986, p.230-239.
- [BARBU92] Barbu, Gh., Modele de simulare cu aplicații în fiabilitate, Editura Tehnică, București, 1992.
- [BEAU78] Beaudry, M.D., Performance related reliability measures for computing systems, IEEE Transactions on Computers, Vol.27, No.6, Jun.1978, p.540-547.
- [BENJ86] Benjamin, R., Analysis of connection survivability in complex strategic communications networks, IEEE Journal on Selected Areas in Communications, Vol.4, No.2, Mar.1986, p.243-253.
- [BENN82] Bennetts, R.G., Analysis of reliability block diagrams by Boolean techniques, IEEE Transactions on Reliability, Vol.31, No.2, Jun.1981, p.159-166.
- [BHAR83] Bharath-Kumar, K., Jaffe, J.M., Routing to multiple destinations in computer networks, IEEE Transactions on Communications, Vol.31, No.3, Mar.1983, p.343-351.

- [BOED93] Boedigheimer, R.A., Kapur, K.C., Involving the customer in the development and evaluation of multistate reliability models, Proceedings of Annual Reliability and Maintainability Symposium, IEEE, 1993, p.250-255.
- [BOED94] Boedigheimer, R.A., Kapur, K.C., Customer-driven reliability models for multistate coherent systems, IEEE Transactions on Reliability, Vol.43, No.1, Mar.1994, p.46-50.
- [BOES86] Boesch, F.T., Synthesis of reliable networks - a survey, IEEE Transactions on Reliability, Vol.35, No.3, Aug.1986, p.240-246.
- [BULKA94] Bulka, D., Bechta Dugan, J., Network s-t reliability bounds using a 2-dimensional reliability polynomial, IEEE Transactions on Reliability, Vol.43, No.1, Mar.1994, p.39-45.
- [BUZA83] Buzacott, J.A., The ordering of terms in cut-based recursive disjoint products, IEEE Transactions on Reliability, Vol.32, No.5, Dec.1983, p.472-474.
- [CARR86] Carrasco, E.H., Colbourn, C.I., Reliability bounds for networks with statistical dependence, The Proceedings of INFOCOM '86, 1986, p.290-292.
- [CATA93] Catania, V., Puliafito, A., Vita, L., A modular network architecture for performance enhancement in extended local area network, IEEE Transactions on Reliability, Vol.42, No.1, Mar.1993, p.50-62.
- [CĂTU89a] Cătuneanu, V.M., Bacivarof, A., Structuri electronice de înaltă fiabilitate, Editura Militară, București, 1989.
- [CĂTU89b] Cătuneanu, V.M., Popențiu, F., Optimizarea fiabilității sistemelor, Editura Academiei, București, 1989.
- [CHAN80] Chan, W.S., Recursive algorithms for computing end-to-end blocking in a network with arbitrary routing plan, IEEE Transactions on Communications, Vol.28, No.2, Feb.1980, p.153-164.
- [CHEN94] Chen, D.-J., Lin, M.-S., On distributed computing systems reliability analysis under program execution constraints, IEEE Transactions on Computers, Vol.43, No.1, Jan.1994, p.87-97.
- [CHER85] Cherkassky, V., Malek, M., Reliability and fail-softness analysis of multistage interconnection networks, IEEE Transactions on Reliability, Vol.34, No.5, Dec.1985, p.524-528.
- [CHIU87] Chiu, G.-M., Raghavendra, C.S., An algorithm for optimal file allocation in distributed computing systems, The Proceedings of INFOCOM '87, 1987, p.72-76.
- [COHEN94] Cohen, R., Segall, A., A new protocol for route discovery in multiple-ring networks: part I - the basic protocol, IEEE Transactions on Communications, Vol.42, No.2/3/4, Feb./Mar./Apr.1994, p.488-490.
- [CUTR94] Cutright, E.D., Johnson, B.W., A simulation-based approach to integrated performance & reliability modeling using VHDL, Proceedings of Annual Reliability and Maintainability Symposium, IEEE, 1994, p.402-408.
- [DALAL87] Dalal, Y., A distributed algorithm for constructing minimal spanning trees, IEEE Transactions on Software Engineering, Vol.13, No.3, Mar.1987, p.398-405.
- [DAS87] Das, C.R., Bhuyan, L.N., Sarma, V.V.S., Effect of maintenance on the dependability and performance of multiprocessor systems, IEEE Transactions on Reliability, Vol.36, No.2, Jun.1987, p.208-215.



- [GARC85] Garcia-Molina, H., Kent, J., Evaluating response time in a faulty distributed computing system, IEEE Transactions on Computers, Vol.34, No.2, Feb.1985, p.101-109.
- [GEBER84] Geber, T. și col., Fiabilitatea, diagnoza și mentenabilitatea sistemelor de calcul, Ed. Tehnică, București, 1984.
- [GIRA83] Girard, A., Quinet, Y., End-to-end blocking for circuit-switched networks: polynomial algorithms for some special cases, IEEE Transactions on Communications, Vol.31, No.12, Dec.1983, p.1269-1272.
- [GITL92] Gitlin, R.D., Hayes, J.F., Weinstein, S.B., Data Communication Principles, Plenum Press, New York, 1992.
- [GÖRKE89] Görke, W., Fehlertolerante Rechensysteme, Oldenbourg Verlag, München, Wien, 1989.
- [GRNA81] Grnarov, A., Gerla, M., Multiterminal reliability analysis of distributed processing systems, The Proceedings of the 1981 International Conference on Parallel Processing, Aug.1981, p.79-86.
- [GROSH89] Grosh, D.L., A Primer of Reliability Theory, John Wiley & Sons, 1989.
- [HARI87] Hariri, S., Raghavendra, C.S., SYREL: A symbolic reliability algorithm based on path and cutset methods, IEEE Transactions on Computers, Vol.36, No.10, Oct.1987, p.1224-1232.
- [HEFF86] Heffes, H., Kumar, A., Incorporating dependent node damage in deterministic connectivity analysis and synthesis of networks, Networks, Vol.16, 1986, p.51-65.
- [HIGA94] Higashiyama, Y., Ariyoshi, H., Shirakawa, I., Ohba, S., A factored reliability formula for directed source-to-all-terminal networks, IEICE Trans. Fundamentals, Vol.E77-A, No.1, Jan.1994, p.134-142.
- [HOHAN82] Hohan, I., Tehnologia și fiabilitatea sistemelor. Ed. Didactică și Pedagogică, București, 1982.
- [HONG93] Hong, J.S., Lie, C.H., Joint reliability-importance of two edges in an undirected network, IEEE Transactions on Reliability, Vol.42, No.1, Mar.1993, p.17-23.
- [HUȚA83] Huțanu, C., Circuite logice și comenzi secvențiale, Ed. Junimea, Iași, 1983.
- [HWANG81] Hwang, C.L., Tillman, F.A., Lee, M.H., System reliability evaluation techniques for complex large systems - a review, IEEE Transactions on Reliability, Vol.30, No.5, Dec.1981, p.416-423.
- [HWANG82] Hwang, K., Chang, T.-P., Combinational reliability analysis of multiprocessor computers, IEEE Transactions on Reliability, Vol.31, No.5, Dec.1982, p.469-473.
- [HWANG84] Hwang, C.L., Lee, H.-B., Tillman, F.A., Lie, C.H., Nonlinear integer goal programming applied to optimal system reliability, IEEE Transactions on Reliability, Vol.33, No.5, Dec.1984, p.431-438.
- [HWANG93] Hwang, K., Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill Inc., 1993.
- [HYUN75] Hyun, K.-N., Reliability optimization by 0-1 programming for a system with several failure modes, IEEE Transactions on Reliability, Vol.24, No.3, Aug.1975, p.206-210.
- [IRANI82] Irani, K.B., Khabbaz, N.G., A methodology for the design of communication networks and the distribution of data in distributed supercomputer systems, IEEE Transactions on Computers, Vol.31, No.5, May.1982, p.419-434.

- [JAIN91] Jain, R., *The Art of Computer Systems Performance Analysis*, John Wiley & Sons, 1991.
- [JAN93] Jan, R.-H., Hwang, F.-J., Cheng, S.-T., Topological optimization of a communication network subject to reliability constraint. *IEEE Transactions on Reliability*, Vol.42, No.1, Mar.1993, p.63-70.
- [JASM85] Jasmon, G.B., Kai, O.S., A new technique in minimal path and cutset evaluation, *IEEE Transactions on Reliability*, Vol.34, No.2, Jun.1985, p.136-143.
- [JASM94] Jasmon, G., Clarification, *IEEE Transactions on Reliability*, Vol.43, No.1, Mar.1994, p.60.
- [JENG88] Jeng, M., Siegel, H.J., Design and analysis of dynamic redundancy networks, *IEEE Transactions on Computers*, Vol.37, No.9, Sept.1988, p.1019-1029.
- [KAKAS92] Kakas, Șt., Fiabilitatea sistemelor multiprocesor reprezentate prin grafuri. O metodă probabilistă de cercetare, *Analele Universității din Oradea, Vol.I, Fascicola Electrotehnică și Energetică*, 1992, p.223-227.
- [KAKAS93] Kakas, Șt., Caracteristici generale de fiabilitate ale sistemelor multiprocesor, *Referat de doctorat*, 1993.
- [KAKAS94a] Kakas, Șt., Analiza fiabilității unei rețele multinivel cu redirișare puternică, *Analele Universității din Oradea, Fascicola Electrotehnică*, 1994, p.527-531.
- [KAKAS94b] Kakas, Șt., Reliability evaluation of a distributed development system, *Proceedings of International Conference on Development and Application Systems D&AS, Suceava, Vol.5, May 1994*, p.197-202.
- [KAKAS94c] Kakas, Șt., Probabilistic problems in the reliability of computer networks solved in the Boolean domain, *Proceedings of The Second Conference on Applied and Industrial Mathematics, Oradea, Aug.1994*, p.41-45.
- [KAKAS94d] Kakas, Șt., Diagnoza sistemelor multiprocesor, *Referat de doctorat*, 1994.
- [KAKAS94e] Kakas, Șt., Evaluarea și creșterea fiabilității sistemelor multiprocesor reprezentate prin grafuri, *Referat de doctorat*, 1994.
- [KAKAS95a] Kakas, Șt., Simplificări în calculul funcției de fiabilitate dintr-o rețea de calculatoare, *Analele Universității din Oradea, Fascicola Electrotehnică*, 1995, p.68-72.
- [KAKAS95b] Kakas, Șt., Importanța fiabilității legăturilor în sinteza rețelelor de calculatoare, *Analele Universității din Oradea, Fascicola Electrotehnică*, 1995, p.73-77.
- [KAKAS95c] Kakas, Șt., Mathematical model for performance assessment of a multiprocessor system. Preprints of 113<sup>th</sup> Pannonian Applied Mathematical Meeting, Bardejovské Kúpele - Slovakia, Oct.1995, p.86-91.
- [KAKAS95d] Kakas, Șt., Studiul performanțelor unui sistem multiprocesor de tip crossbar, *Revista Română de Automatică, Nr.4 1995*, p.1-5.
- [KAKAS96a] Kakas, Șt., Performance model for a multiprocessor system validated by computer simulation, *Proceedings of International Computer Science Conference microCAD'96, Miskolc - Hungary, Feb.1996*, p.67-70.
- [KAKAS96b] Kakas, Șt., Performance and reliability measures for multiprocessor systems, *Proceedings of Workshop, Kosice - Slovakia, Apr.1996*, p.14-21.
- [KAKAS96c] Kakas, Șt., The influence of failures of interconnecting elements on the performance of a multiprocessor system, *Automatic Control and Testing Conference A'96-THETA 10, Cluj-Napoca, May 1996, Vol.3*, p.197-200.

- [KAKAS96d] Kakas, Şt., Integrated performance and reliability model for a multiprocessor system, International Symposium on System Theory SINTES 8 (proceedings), Craiova, Jun.1996, p.113-118.
- [KINI82] Kini, V., Siewiorek, D.P., Automatic generation of symbolic reliability functions for processor-memory-switch structures, IEEE Transactions on Computers, Vol.31, No.8, Aug.1982, p.752-771.
- [KRUS88] Kruskal, C.P., Smith, C.H., Definitions of granularity (in High Performance Computer Systems, editor E. Gelenbe), North Holland, 1988.
- [KUBAT86] Kubat, P., Reliability analysis for integrated networks with application to burst switching, IEEE Transactions on Communications, Vol.34, No.6, Jun.1986, p.564-568.
- [KUMAR88a] Kumar, A., Rai, S., Agrawal, D.P., Reliability evaluation algorithms for distributed systems, The Proceedings of INFOCOM '88, 1988, p.851-860.
- [KUMAR88b] Kumar, A., Rai, S., Agrawal, D.P., On computer communication network reliability under program execution constraints, IEEE Transactions on Selected Areas in Communications, Vol.6, No.8, Oct.1988, p.1393-1400.
- [KUO87] Kuo, W., Lin, H.-H., Xu, Z., Zhang, W., Reliability optimization with the Lagrange-multiplier and branch-and-bound technique, IEEE Transactions on Reliability, Vol.36, No.5, Dec.1987, p.624-630.
- [LAM86] Lam, Y., F., Li, V.O.K., Reliability modeling and analysis of communication networks with dependent failures, IEEE Transactions on Communications, Vol.34, No.1, Jan.1986, p.82-84.
- [LAWR75] Lawrie, D.H., Access and alignment of data in array processor, IEEE Transactions on Computers, Vol.24, No.12, Dec.1975, p.1145-1155.
- [LEE76] Lee, A.M., Teoria aşteptării cu aplicații, Ed. Tehnică, București, 1976.
- [LEE80] Lee, S.H., Reliability evaluation of a flow network, IEEE Transactions on Reliability, Vol.29, No.1, Apr.1980, p.24-26.
- [LEE92] Lee, C.-H., Lee, D., Kim, M., Optimal task assignment in linear array networks, IEEE Transactions on Computers, Vol.41, No.7, Jul.1992, p.877-880.
- [LEVI94] Levi, S.-T., Agrawala, A.K., Fault Tolerant System Design, McGraw-Hill Inc., 1994.
- [LIN78] Lin, P.M., Leon, B.J., Stewart, C.R., Analysis of circuit-switched networks employing originating-office control with spill-forward, IEEE Transactions on Communications, Vol.26, No.6, Jun.1978, p.754-765.
- [LIN93] Lin, J.Y., Donaghey, C.E., A Monte Carlo simulation to determine minimal cut sets and system reliability, Proceedings of Annual Reliability and Maintainability Symposium, IEEE, 1993, p.246-249.
- [LOCKS87] Locks, M.O., A minimizing algorithm for the sum of disjoint products, IEEE Transactions on Reliability, Vol.36, No.4, Oct.1987, p.445-453.
- [LOPEZ92] Lopez-Benitez, N., Dependability analysis of distributed computing systems using stochastic Petri nets, Proceedings of the 11<sup>th</sup> Symposium on Reliable Distributed Systems, IEEE, 1992, p.85-92.
- [MACA88] Macaluso, J.J., Das, C.R., Lin, W., A reliability predictor for MIN-connected multiprocessor systems, Proceedings of the 1988 International Conference on Parallel Processing, IEEE, 1988, p.392-399.
- [MAYE72] Mayeda, W., Graph Theory, John Wiley and Sons, New York, 1972.

- [MELUS88] Melus, J.L., Sanvicente, E., Magrina, J., A memory interference model for multiprocessors (in High Performance Computer Systems, editor E. Gelenbe), North Holland, 1988.
- [MEYER80] Meyer, J.F., On evaluating the performability of degradable computing systems, IEEE Transactions on Computers, Vol.29, No.8, Aug.1980, p.720-731.
- [MIHOC67] Mihoc, Gh., Ciucu, G., Introducere în teoria așteptării, Ed. Tehnică, București, 1967.
- [MIHOC73] Mihoc, Gh., Ciucu, G., Muja, A., Modele matematice ale așteptării, Ed. Academiei, București, 1973.
- [MIL86] MIL-HDBK-217: Reliability Prediction of Electronic Equipment, Oct.1986.
- [MUNT77] Muntean, I., Sinteza automatelor finite, Ed. Tehnică, București, 1977.
- [MURR93] Murray, K., Kershenbaum, A., Schooman, M.L., Communication network reliability analysis approximations and bounds, Proceedings of Annual Reliability and Maintainability Symposium, IEEE, 1993, p.268-275.
- [NAKA81] Nakazawa, H., Decomposition method for computing the reliability of complex networks, IEEE Transactions on Reliability, Vol.30, No.3, Aug.1981, p.289-292.
- [NÉME93] Németh, G., Horváth, L., Számítógép architektúrák, Akadémiai Kiadó, Budapest, 1993.
- [NEVA88] Nevalainen, S.J., Minimization of Boolean sum of products with applications to the computation of reliability of general networks, Doctoral thesis, VTT Espoo - Finland, 1988.
- [PAGE94] Page, L.B., Perry, J.F., Reliability polynomials and link importance in networks, IEEE Transactions on Reliability, Vol.43, No.1, Mar.1994, p.51-58.
- [PĂTR94] Pătrășcoiu, O., Marian, G., Mitroi, N., Elemente de grafuri și combinatorică. Metode, algoritmi și programe, Editura ALL, București, 1994.
- [PĂUN93] Păunescu, Fl., Goleșteanu D.P., Sisteme cu prelucrare distribuită și aplicațiile lor, Editura Tehnică, București, 1993.
- [POLI86] Politof, T., Satyanarayana, A., Efficient algorithms for reliability analysis of planar networks - a survey, IEEE Transactions on Reliability, Vol.35, No.3, Aug.1986, p.252-259.
- [PRAD86] Pradhan, D.K., Fault-Tolerant Computing: Theory and Techniques, Prentice-Hall, New Jersey, 1986.
- [PRAS86] Prasanna Kumar, V.K., Hariri, S., Raghavendra, C.S., Distributed program reliability analysis, IEEE Transactions on Software Engineering, Vol.12, No.1, Jan.1986, p.42-50.
- [PRAS94] Prasad, V.C., Sankar, V., Prakasa Rao, K.S., Comment on "Cut set analysis of networks using basic minimal paths and network decomposition, IEEE Transactions on Reliability, Vol.43, No.1, Mar.1994, p.59-60.
- [PROV84] Provan, J.S., Ball, M.O., Computing network reliability in time polynomial in the number of cuts, Operations Research, Vol.32, No.3, May-Jun.1984, p.516-526.
- [PROV86] Provan, J.S., Bounds on the reliability of networks, IEEE Transactions on Reliability, Vol.35, No.3, Aug.1986, p.260-268.

- [RAGH85] Raghavendra, C.S., Hariri, S., Reliability optimization in the design of distributed systems, IEEE Transactions on Software Engineering, Vol.11. No.10, Oct.1985, p.1184-1193.
- [RAI78] Rai, S., Aggarwal, K.K., An efficient method for reliability evaluation of a general network, IEEE Transactions on Reliability, Vol.27, No.3, Aug.1978, p.206-211.
- [RAI80] Rai, S., Aggarwal, K.K., On complementation of pathsets and cutsets, IEEE Transactions on Reliability, Vol.29, No.2, Jun.1980, p.139-140.
- [RAI86a] Rai, S., Kumar, A., Prasad, E.V., Computing terminal reliability of computer network, Reliability Engineering, Vol.16, Jun.1986, p.109-119.
- [RAI86b] Rai, S., Kumar, A., Prasad, E.V., Computing the performance index of a computer network, Reliability Engineering, Vol.16, Jun.1986, p.153-161.
- [RAI87] Rai, S., Sarje A.K., Prasad, E.V., Kumar, A., Two recursive algorithms for computing the reliability of k-out-of-n systems, IEEE Transactions on Reliability, Vol.36, No.2, Jun.1987, p.261-265.
- [RAI90a] Rai, S., Agrawal, D.P., Advances in Distributed System Reliability, IEEE Computer Society Press, Los Alamitos CA, 1990.
- [RAI90b] Rai, S., Agrawal, D.P., Distributed Computing Network Reliability, IEEE Computer Society Press, Los Alamitos CA, 1990.
- [RUEG86] Rueger, W.J., Reliability analysis of networks with capacity-constraints and failures at branches & nodes, IEEE Transactions on Reliability, Vol.35, No.5, Dec.1986, p.523-528.
- [SAHN87] Sahner, R.A., Trivedi, K.S., Reliability modeling using SHARPE, IEEE Transactions on Reliability, Vol.36, No.2, Jun.1987, p.186-193.
- [SANK93] Sankar, V., Prasad, V.C., Prakasa Rao, K.S., Comment on: enumeration of all minimal cutsets for a node pair in a graph, IEEE Transactions on Reliability, Vol.42, No.1, Mar.1993, p.44-45.
- [SATY81] Satyanarayana, A., Hagstrom, J.N., A new algorithm for the reliability analysis of multi-terminal networks, IEEE Transactions on Reliability, Vol.30, No.4, Oct.1981, p.325-334.
- [SATY85] Satyanarayana, A., Wood, R.K., A linear algorithm for computing K-terminal reliability in series-parallel networks, SIAM Journal of Computing, Vol.14, No.4, Nov.1985, p.818-832.
- [SCHE84] Scheuermann, P., Wu, G., Heuristic algorithms for broadcasting in point-to-point computer networks. IEEE Transactions on Computers, Vol.33, No.9, Sep.1984, p.804-811.
- [SCHO86] Schoen, O., On a class of integrated performance-reliability models based on queuing networks, The Proceedings of the International Symposium on Fault-Tolerant Computing Systems, IEEE, 1986, p.90-95.
- [SHAN82] Shanthikumar, J.G., Recursive algorithm to evaluate the reliability of a consecutive-k-out-of-n:F system, IEEE Transactions on Reliability, Vol.31, No.5, Dec.1982, p.442-443.
- [SHAR93] Sharma, T.C., Bazovsky Sr., I., Reliability analysis of large system by Markov techniques, Proceedings of Annual Reliability and Maintainability Symposium, IEEE, 1993, p.260-267.
- [SHATZ92] Shatz, S.M., Wang, J-P., Goto, M., Task allocation for maximizing reliability of distributed computer systems, IEEE Transactions on Computers, Vol.41, No.9, Sep.1992, p.1156-1168.

- [SHOO92] Shooman, A.M., Kershenbaum, A., Methods for communication-network reliability analysis: probabilistic graph reduction, Proceedings of Annual Reliability and Maintainability Symposium, IEEE, 1992, p.441-448.
- [SIMM94] Simmons, J.M., A strategy for designing error detection schemes for general data networks, IEEE Network, Jul./Aug.1994, p.41-48.
- [SMITH88] Smith, R.M., Trivedi, K.S., Ramesh, A.V., Performability analysis: measures, an algorithm and a case study, IEEE Transactions on Computers, Vol.37, No.4, Apr.1988, p.406-417.
- [SOH93] Soh, S., Rai, S., Experimental results on preprocessing of path/cut terms in sum of disjoint products technique, IEEE Transactions on Reliability, Vol.42, No.1, Mar.1993, p.24-33.
- [SPRA81] Spragins, J.D., Markov, J.D., Doss, M.W., Mitchell, S.A., Squire, D.C., Communication network availability predictions based on measurement data, IEEE Transactions on Communications, Vol.29, No.10, Oct.1981, p.1482-1491.
- [SPRA86] Spragins, J.D., Sinclair, J.C., Kaug, Y.J., Jafari, H., Current telecommunication network reliability models: a critical assessment, IEEE Journal on Selected Areas in Communications, Vol.4, No.7, Oct.1986, p.1186-1173.
- [SZIT90] Szittyá, O., Logikai rendszerek és szekvenciális automaták, Budapesti Műszaki Egyetem, 1990.
- [TABAK90] Tabak, D., Multiprocessors, Prentice-Hall International, 1990.
- [TAYL94] Taylor, S.J.E., Performance evaluation of parallel systems using simulation, Proceedings of The Eighth Symposium on Microcomputer and Microprocessor Applications, Budapest - Hungary, Vol.I, Oct.1994, p.220-229.
- [TÂRC89] Târcolea, C., Filipoiu, A., Bontaş, S., Tehnici actuale în teoria fiabilităţii, Ed. Ştiinţifică şi Enciclopedică, Bucureşti, 1989.
- [TEWA92] Tewari, R., Adam, N., Distributed file allocation with consistency constraints, Proceedings of the Conference on Distributed Computing Systems, IEEE, 1992, p.408-415.
- [TORR83] Torrey, J., A pruned tree approach to reliability computation, IEEE Transactions on Reliability, Vol.32, No.2, Jun.1983, p.170-174.
- [TZENG88] Tzeng, N.-F., Yew, P.-C., Zhn, C.-Q., Realizing fault-tolerant interconnection networks via chaining, IEEE Transactions on Computers, Vol.37, No.4, Apr.1988, p.458-462.
- [VÁDU77] Văduva, I., Modele de simulare cu calculatorul, Ed. Tehnică, Bucureşti, 1977.
- [VISW87] Viswanadhan, N., Sarma, V.S.S., Singh, M.S., Reliability of Computer and Control Systems, North Holland - Elsevier, 1987.
- [VEER91] Veeraraghavan, M., Trivedi, K., An improved algorithm for symbolic reliability analysis, IEEE Transactions on Reliability, Vol.40, No.3, Aug.1991, p.347-358.
- [VEER93] Veeraraghavan, M., Trivedi, K.S., An approach for combinatorial performance and availability analysis, Proceedings of the 12<sup>th</sup> Symposium on Reliable Distributed Systems, IEEE, 1993, p.24-33.
- [VEER94] Veeraraghavan, M., Trivedi, K.S., A combinatorial algorithm for performance and reliability analysis using multistate models, IEEE Transactions on Computers, Vol.43, No.2, Feb.1994, p.229-234.

- [VLĂD82] Vlăduțiu, M., Tehnologie de ramură și fiabilitate (curs), I.P."Traian Vuia" Timișoara, 1982.
- [VLĂD86] Vlăduțiu, M., Tehnica testării echipamentelor de calcul și evaluarea performanțelor (curs), I.P."Traian Vuia" Timișoara, 1986.
- [VLĂD89] Vlăduțiu, M., Crișan, M., Tehnica testării echipamentelor automate de prelucrare a datelor, Ed. Facla Timișoara, 1989.
- [WILK72] Wilkow, R., Analysis and design of reliable computer networks, IEEE Transactions on Communications, Vol.20, No.6, Jun.1972, p.660-678.
- [WILK91] Wilkinson, B., Computer Architecture: Design and Performance, Prentice Hall, New York, 1991.
- [WILS90] Wilson, J.M., An improved minimizing algorithm for sum of disjoint products, IEEE Transactions on Reliability, Vol.39, No.1, Apr.1990, p.42-45.
- [WITT81] Wittie, L.D., Communication structures for large networks of microcomputers, IEEE Transactions on Computers, Vol.30, No.4, Apr.1981, p.246-273.
- [WOOD85] Wood, A.P., Multistate block diagrams and fault trees, IEEE Transactions on Reliability, Vol.34, No.3, Aug.1985. p.236-240.
- [YU94] Yu, K., Koren, I., Guo, Y., Generalized multistate monotone coherent systems, IEEE Transactions on Reliability, Vol.43, No.2, Jun.1994, p.242-250.