Metrici de complexitate software bazate pe dependentele instructiunilor

Prof.dr.Ion IVAN, A.S.E. Bucuresti
Alexandra KARADIMOU, Adrian LICURICEANU, Gheorghe LUPU

Se definesc metrici bazate pe dependentele dintre instructiuni si se stabilesc proprietatile indicatorilor construiti. Se considera un lot de programe si se fac determinari. Se identifica legatura dintre metricile de complexitate Halstead, McCabe, LIL si metrica bazata pe dependenta instructiunilor.
***Cuvinte cheie**: complexitate software, metrici software, dependenta instructiunilor.*

1. Introducere

Complexitatea software este o caracteristica de calitate care se regaseste în calculul devizelor pentru dezvoltarea software si pentru corectarea indicatorului de productivitate a programatorilor.

Metricile software s-au definit pe baza a trei surse: metrici bazate pe textul sursa, metrici bazate pe graful asociat programului, metrici de comportament care înregistreaza niveluri ale parametrilor în timpul executiei programului.

În [Hals1] este definita metrica Halstead prin indicatorii:

C = complexitatea programului
E = efortul de programare
V = volumul programului
L = nivelul programului

unde:

$$C = \sum_{i=1}^{6} n_i \log_2 n_i$$

sau, considerând $h_1 = \mathrm{n}_1 + \mathrm{n}_2$ si $h_2 = \mathrm{n}_3 + \mathrm{n}_4 + \mathrm{n}_5 + \mathrm{n}_5$ putem scrie:

$$C = h_1 \log_2 h_1 + h_2 \log_2 h_2$$

$$V = N \log_2 \sum_{i=1}^{6} n_i, N = \sum_{i=1}^{6} n_i^*,$$

n_i^* având aceeasi semnificatie ca si n_i, dar contorizeaza totalurile, nu numai pe cele distincte.

$$L = \frac{V^*}{V}$$

V^* este volumul minim al programului, care este calculat din numarul minim de parametrii I/O necesar pentru a specifica operatia unui algoritm si returul rezultatului, având expresia:

$$V^* = h^* \log_2 h^*, h^* = h^*_2 + 2,$$

iar h_2^* este numarul de parametri de I/O folositi în apelul programului.

$$E = \frac{V}{L}$$

cu:

n_1 – numarul de tipuri fundamentale de date care apar distinct în program
n_2 – numarul de tipuri derivate de date care apar distinct în program
n_3 – numarul de instructiuni distincte utilizate de programator
n_4 – numarul de operanzi distincti care apar în program
n_5 – numarul de operatori distincti pentru referire care apar în program
n_6 – numarul de functii distincte apelate

În [McCabe1] este definita metrica bazata pe structura de graf asociat programului, McCabe, astfel., C = n - m + 2, unde n este numarul de arce, iar m numarul de noduri.

În [IvArh1] si [IvArh2] sunt definiti indicatorii de comportament ai programului: durata de compilare, durata de executie, durata medie tranzactie, gradul de stabilitate, durata medie corectie, care necesita efectuarea de înregistrari sub forma seriilor de timp.

Aceste metrici trateaza programele ca sisteme formate din instructiuni pentru care nu prezinta importanta modul în care se succed. Aceasta ipoteza nu este conforma

cu realitatea. Pozitionarea instructiunilor în program are o logica, data de algoritmii implementati, de ordinea în care se efectueaza initializarile, evaluarile de expresii si afisarea de rezultate. Prezinta importanta daca se memoreaza sau nu rezultatele.
În continuare se prezinta metrici care iau în considerare dependenta instructiunilor.

2. Matricea de precedente

Un program P este format din instructiunile I_1, $I_{2, \dots,}$ I_n. Metrica McCabe ia în considerare graful G(V, A) unde V este multimea nodurilor si A este multimea arcelor.
Se considera programul urmator:

```
void main ()
{                                   I1
int a, b, c, d, e, x;
a= 3;                               I2
b = 5;                              I3
c =8;                               I4
b = 7;                              I5
scanf("%d", &x);                    I6
if (x>0)                            I7
        e = a + b;                  I8
else
        e = c + d;                  I9
printf("%d",e);                     I10
}                                   I11
```

Programului i se asociaza graful G din figura 1.
Matricea P, a precedentelor, asociata grafului G, este :

$$P = \begin{array}{c} \\ I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \\ I_7 \\ I_8 \\ I_9 \\ I_{10} \\ I_{11} \end{array} \begin{array}{c} \begin{array}{ccccccccccc} I_1 & I_2 & I_3 & I_4 & I_5 & I_6 & I_7 & I_8 & I_9 & I_{10} & I_{11} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

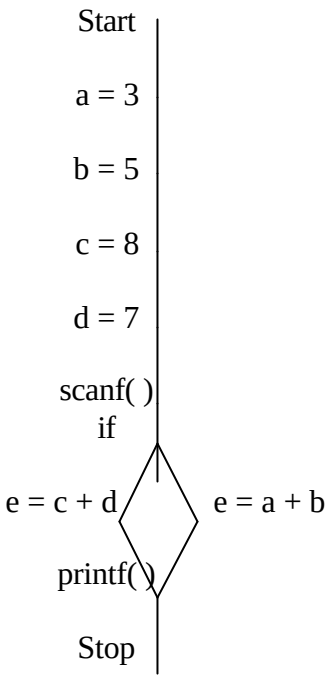

Fig . 1 Graful asociat programului.

Precedenta este rezultatul dispunerii instructiunilor în program în asa fel încât acestea sa conduca la obtinerea de rezultate corecte. O problema se obtine prin scrierea a N programe diferite, care însa implementeaza acelasi algoritm. Diferentele care apar sunt determinate de: limbajul utilizat, structurile de date definite si referite, structurile de control folosite, gradul de generalitate, dimensiunea problemelor care se rezolva, precizia rezultatelor, optiunile privind datele de intrare si structura rezultatelor obtinute, nivelul de reutilizabilitate a procedurilor, a tipurilor de date, a claselor.
Matricele de precedenta difera de la program la program, chiar daca este vorba de rezolvarea aceleiasi probleme. Pentru ca instructiunile, succesiunea lor determina grafuri diferite. În [Ivan1] este definita metrica bazata pe matricea de precedenta, cu luarea în considerare a lungimilor arcelor, determinate de diferenta în modul a pozitiei instructiunilor executate consecutiv. Astfel metrica LIL1 se defineste prin:

$$LIL1 = \sum_{ii} \sum_{j} |i - j|,\ i \neq j \ si\ (i, j) \in A\ ,$$

iar metrica LIL2 se defineste prin:

$$LIL2 = \sum_{ii} \sum_{j} (i-j)^2,\ i \neq j\ si\ (i,j) \in A$$

Experimentele evidentiaza diferentele care apar între programele în care lungimea arcelor este mare, ceea ce metrica McCabe nu reflecta.

Software-ul elaborat pentru implementarea metricilor LIL1, LIL2 precum si a formelor normate: LIL1N=LIL1/max(LIL1), LIL2N=LIL2/max(LIL2), conduce la evaluarea coeficientilor de corelatie ale caror niveluri sunt cuprinse între 0.70 si 0.90, ceea ce exprima posibilitatea utilizarii oricarei dintre metrici, rezultatele fiind asemanatoare.

3. Matricea dependentelor

Se considera programul P, format din instructiunile $I_1, I_2, ..., I_n$. Instructiunea I_k depinde de instructiunile $I_{k-1}, I_{k-2}, ..., I_{k-r}$ daca variabilele initializate sau modificate de acestea intervin activ în expresii sau determina controlul executiei I_k.

Matricea B, având n linii si n coloane, a dependentelor are elementele:

$$b_{ij} = \begin{cases} 0 \text{ daca instr. i nu depinde de instr. j} \\ 1 \text{ daca instr. i depinde de instr. j} \end{cases}$$

Se considera b_{ii}=1, instructiunea depinzând de ea insasi.

Se considera programul P1:

```
void main( )
{   int a, b, c, d;
    a = 1;                  // I1
    b = 2;                  // I2
    c = 7;                  // I3
    d = 114;                // I4
}
```

Programului P1 i se asociaza matricea dependentelor:

	I_1	I_2	I_3	I_4
I_1	1	0	0	0
I_2	0	1	0	0
I_3	0	0	1	0
I_4	0	0	0	1

Se observa ca instructiunile programului P1 sunt independente una de celelalte si li se asociaza matricea unitate a dependentelor.

Programului P2 :

```
void main( )
{
int a, b, c, d;,e
a = 4;                  // I1
b = a+5;                // I2
c = b+120;              // I3
d = b+a+c;              // I4
e = c+d;                // I5
printf (" \n %d %d %d %d %d", a,b,c,d,e);
                        // I6
}
```

i se asociaza matricea dependentelor:

	I_1	I_2	I_3	I_4	I_5	I_6
I_1	1	0	0	0	0	0
I_2	1	1	0	0	0	0
I_3	0	1	1	0	0	0
I_4	1	1	1	1	0	0
I_5	0	0	1	1	1	0
I_6	1	1	1	1	1	1

Programului P3 :

```
void    main ( )
{
int a,b,c,d,e,i;
scanf( "%d",&a);            // I1
if (a>5) {                  // I2
        scanf("%d",&b);     // I3
        d = a+b;            // I4
        }
else {
        scanf("%d",&c);     // I5
        d = a + c ;         // I6
        }
e = d ++;                   // I7
for(i =0 ;i<7;i++)          // I8
        e += a;             // I9
printf("%d",e);             // I10
}
```

i se asociaza matricea dependentelor:

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
I_1	1	0	0	0	0	0	0	0	0	0
I_2	1	1	0	0	0	0	0	0	0	0
I_3	0	1	1	0	0	0	0	0	0	0
I_4	0	1	1	1	0	0	0	0	0	0
I_5	0	1	0	0	1	0	0	0	0	0
I_6	1	1	0	0	1	1	0	0	0	0
I_7	0	0	0	1	0	1	1	0	0	0
I_8	0	0	0	0	0	0	0	1	0	0
I_9	1	0	0	0	0	0	1	1	1	0
I_{10}	0	0	0	0	0	0	0	0	1	1

4. Indicatorul de dependente

Se considera matricea dependentelor B având elementele $b_{ij} \in \{0,1\}$. Se calculeaza complexitatea:

$$C_d = \sum_{i=1}^{n}((\sum_{j=1}^{n} b_{ij})\log_2(\sum_{j=1}^{n} b_{ij}))$$

bazata pe dependente. C_d apartine intervalului $[0, q]$ unde $q = \sum_{i=1}^{n} i \log_2 i$ pentru ca se lucreaza cu matricea inferior triunghiulara.

Normarea indicatorului bazat pe dependente se va realiza astfel:

$$\overline{C_d} = \frac{\sum_{i=1}^{n}(\sum_{j=1}^{n} b_{ij}\log_2(\sum_{j=1}^{n} b_{ij}))}{\sum_{i=1}^{n} i\log_2 i}.$$

Matricea dependentelor maxime pentru un program cu 5 instructiuni este:

	I_1	I_2	I_3	I_4	I_5
I_1	1	0	0	0	0
I_2	1	1	0	0	0
I_3	1	1	1	0	0
I_4	1	1	1	1	0
I_5	1	1	1	1	1

Daca programul are toate instructiunile de atribuire independente între ele sau contin exclusiv afisarea unor texte, atunci matricea dependentelor este o matrice unitate si C_d, $\overline{C_d}$ sunt 0.

Instructiunile se grupeaza în doua categorii în functie de dependenta: instructiuni slab dependente si instructiuni puternic dependente.

Se noteaza $b_i = \sum_{j=1}^{n} b_{ij}$. Se alege elementul maxim din sirul $b_1, b_2, \ldots, b_n$:

$$b_{max} = \max_{i=1,n}\{b_i\}$$

Se calculeaza valoarea $\Delta = (b_{max} - 1)/2$.

Fie:

$$M = \sum_{\substack{i=1 \\ 1 \le b_i \le \Delta+1}}^{n} b_i$$

$$N = \sum_{\substack{i=1 \\ \Delta+1 < b_i \le b_{max}}}^{n} b_i$$

Complexitatea gradata a programului este data de relatia :

$$Cg = M\log_2 M + N\log_2 N$$

$$\overline{Cg} = \frac{M\log_2 M + N\log_2 N}{(M+N)\log_2(M+N)}$$

Se considera un program P4 având matricea dependentelor:

$$\begin{pmatrix} 1 & & & & & & \\ 1 & 1 & & & & & \\ 1 & 0 & 1 & & & & \\ 0 & 1 & 1 & 1 & & & \\ 1 & 0 & 0 & 1 & 1 & & \\ 1 & 0 & 1 & 1 & 1 & 1 & \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Sirul b_i este format din elementele: {1, 2, 2, 3, 3, 5, 4}, $b_{max} = 5$, $\Delta = 2$, M=1+2+2+3+3 = 11, N = 5 + 4 = 9.

$$C_g = 11\log_2 11 + 9\log_2 9 = 66.58$$

$$\overline{C_g} = \frac{11\log_2 11 + 9\log_2 9}{20\log_2 20} = 0.77025$$

Complexitatea calitativa C_a opereaza cu numarul de componente apartinând intervalelor [1, Δ+1] si (Δ+1, b_{max}].

Fie M numarul de elemente din sirul $b_1, b_2, \ldots, b_n$ pentru care $b_i \in$ [1, Δ+1] si N numarul de elemente din sirul

$b_i \in (\Delta+1, b_{max}]$.

Astfel:

$$Ca = M1\log_2 M1 + N1\log_2 N1$$

$$\overline{Ca} = \frac{M1\log_2 M1 + N1\log_2 N1}{n\log_2 n}$$

Pentru programul P4:

M=5, numarul de elemente din sirul b_1, b_2, ..., b_n pentru care $b_i \in [1, \Delta+1]$, iar N=2, numarul de elemente din sirul

$b_i \in (\Delta+1, b_{max}]$.

$$Ca = 5\log_2 5 + 2\log_2 2 = 13,6$$

$$\overline{Ca} = \frac{5\log_2 5 + 2\log_2 2}{7(\lg_2 7 - 1)} = 0,69$$

5. Rezultate experimentale

Se considera lotul de programe cu caracteristicile date în tabelul nr.1:

Tabel nr. 1

Cod program	Tip problema	Nr instructiuni	Lungime fisier normalizata
Cmmdc.cpp	CMMDC	33	0,18
Concat.cpp	Concatenare siruri	25	0,14
1.cpp	Ordonare si inserare	41	0,34
2.cpp	Creare matrice triunghiulara	18	0,15
3.cpp	Calcul salariu	30	0,26
4.cpp	Inversare sir	14	0,098
5.cpp	Compresie pe subsiruri	34	0,31
Hotteling.cpp	Metoda Hotteling	144	0,99
Polinom.cpp	Valoare polinom	51	0,42
Pro1.cpp	Ridicari la putere	22	0,18
Pro2.cpp	Numarare spatii	16	0,102
Sircar.cpp	Inserare caracter	27	0,3
Sistprosc.cpp	Rezolvarea sistemelor prost conditionate	139	1

Se calculeaza complexitatile Halstead, McCabe, LIL, precum si metrica bazate pe dependente software C_d, iar rezultatele analizei sunt prezentate în tabelul 2:

Tabel nr. 2

Cod program	Halstead	McCabe	LIL	C_d
Cmmdc.cpp	63.87	6	49	45.02
Concat.cpp	58.89	3	30	36.26
1.cpp	147.23	10	62	68.04
2.cpp	113.73	6	32	26
3.cpp	39.12	1	29	1
4.cpp	51.28	4	20	2
5.cpp	75.91	9	63	16.75
Hotteling.cpp	294.57	46	399	365.92
Polinom.cpp	106.67	12	107	28
Pro1.cpp	65.58	5	34	20.26
Pro2.cpp	33.87	3	21	12
Sircar.cpp	86.9	5	39	23.51
Sistprosc.cpp	448.29	34	302	401.89

Se realizeaza o ordonare în functie de complexitatea Halstead. Indexul H = {1, 2, …, n} corespunde programelor ordonate descrescator dupa nivelul acestei complexitati. Se face si o ordonare în functie de nivelul complexitatii McCabe, rezultând indexul Mc={1, 2, …, n}. Tot asa se ordoneaza dupa toti indicatorii din tabelul 2, obtinându-se tabelul 3, în care ordinea rândurilor este data de prima ordonare, cea dupa nivelul complexitatii Halstead:

Tabel nr. 3

Cod program	**Indice Halstead**	**Indice McCabe**	**Indice LIL**	**Indice C_d**
Sistprosc.cpp	1	2	2	1
Hotteling.cpp	2	1	1	2
1.cpp	3	4	5	3
2.cpp	4	6	9	7
Polinom.cpp	5	3	3	6
Sircar.cpp	6	8	7	8
5.cpp	7	5	4	10
Pro1.cpp	8	9	8	9
Cmmdc.cpp	9	7	6	4
Concat.cpp	10	11	10	5
4.cpp	11	10	13	12
3.cpp	12	13	11	13
Pro2.cpp	13	12	12	11

Se considera astfel indexul Halstead ca fiind referinta. Se calculeaza indicatorul:

$$P(H,X) = 1 - \frac{\sum_{i=1}^{n}(H_i - X_i)^2}{2\sum_{i=1}^{[n/2]}(2i)^2}$$ unde:

H – metrica Halstead
Hi – numarul de ordine al programului Pi în functie de nivelul complexitatii Halstead
X – una din complexitatile calculate, mai putin Halstead
Xi – numarul de ordine al programului Pi în functie de nivelul complexitatii X
[ ] – functia parte întreaga
Se obtin urmatoarele rezultate:
P(Halstead, McCabe) = 0.97
P(Halstead, LIL) = 0.91
P(Halstead, Cd) = 0.93
Considerându-se indexul McCabe referinta se obtine tabelul nr.4:

Tabel nr. 4

Cod program	Indice Halstead	Indice McCabe	Indice LIL	Indice C_d
Hotteling.cpp	2	1	1	2
Sistprosc.cpp	1	2	2	1
Polinom.cpp	5	3	3	6
1.cpp	3	4	5	3
5.cpp	7	5	4	10
2.cpp	4	6	9	7
Cmmdc.cpp	9	7	6	4
Sircar.cpp	6	8	7	8
Pro1.cpp	8	9	8	9
4.cpp	11	10	13	12
Concat.cpp	10	11	10	5
Pro2.cpp	13	12	12	11
3.cpp	12	13	11	13

Se calculeaza indicatorul:

$$P(McC, X) = 1 - \frac{\sum_{i=1}^{n} (McC_i - X_i)^2}{2\sum_{i=1}^{[n/2]} (i-1)^2}$$

unde:
McC – metrica McCabe
McCi – numarul de ordine al programului Pi în functie de nivelul complexitatii McCabe
X – una din complexitatile calculate, mai putin McCabe
Xi – numarul de ordine al programului Pi în functie de nivelul complexitatii X
[ ] – functia parte întreaga
Se obtin urmatoarele rezultate:
P(McCabe, Halstead) = 0.97
P(McCabe, LIL) = 0.96
P(McCabe, Cd) = 0.84

Considerându-se indexul LIL referinta se obtine tabelul nr.5:

Tabel nr. 5

Cod program	Indice Halstead	Indice McCabe	Indice LIL	Indice C_d
Hotteling.cpp	2	1	1	2
Sistprosc.cpp	1	2	2	1
Polinom.cpp	5	3	3	6
5.cpp	7	5	4	10
1.cpp	3	4	5	3
Cmmdc.cpp	9	7	6	4
Sircar.cpp	6	8	7	8
Pro1.cpp	8	9	8	9
2.cpp	4	6	9	7
Concat.cpp	10	11	10	5
3.cpp	12	13	11	13
Pro2.cpp	13	12	12	11
4.cpp	11	10	13	12

Se calculeaza indicatorul:

$$P(L, X) = 1 - \frac{\sum_{i=1}^{n} (L_i - X_i)^2}{2\sum_{i=1}^{[n/2]} (2i)^2}$$

unde:
L – metrica LIL
Li – numarul de ordine al programului Pi în functie de nivelul complexitatii LIL
X – una din complexitatile calculate, mai putin LIL
Xi – numarul de ordine al programului Pi în functie de nivelul complexitatii X
[ ] – functia parte întreaga
Se obtin urmatoarele rezultate:
P(LIL, Halstead) = 0.91
P(LIL, McCabe)= 0.96
P(LIL, Cd)= 0.88

Considerându-se indexul C_d referinta se obtine tabelul nr.6:

Tabel nr. 6

Cod program	Indice Halstead	Indice McCabe	Indice LIL	Indice C_d
Sistprosc.cpp	1	2	2	1
Hotteling.cpp	2	1	1	2
1.cpp	3	4	5	3
Cmmdc.cpp	9	7	6	4
Concat.cpp	10	11	10	5

Polinom.cpp	5	3	3	6
2.cpp	4	6	9	7
Sircar.cpp	6	8	7	8
Pro1.cpp	8	9	8	9
5.cpp	7	5	4	10
Pro2.cpp	13	12	12	11
4.cpp	11	10	13	12
3.cpp	12	13	11	13

Se calculeaza indicatorul:

$$P(Cd, X) = 1 - \frac{\sum_{i=1}^{n} (Cd_i - X_i)^2}{2\sum_{i=1}^{[n/2]} (2i)^2}$$

unde:
Cd – metrica Cd
Cdi – numarul de ordine al programului Pi în functie de nivelul complexitatii Cd
X – una din complexitatile calculate, mai putin Cd
Xi – numarul de ordine al programului Pi în functie de nivelul complexitatii X
[ ] – functia parte întreaga
Se obtin urmatoarele rezultate:
P(Cd, Halstead) = 0.93
P(Cd, McCabe) = 0.84
P(Cd, LIL) = 0.88
Se considera intervalul [0,1] divizat în 3 subintervale, [0, 0.50), [0.50, 0.75), [0.75, 1] corespunzând calificativului slab, bun si foarte bun.
Pentru metricile de complexitate considerate se obtin valorile din tabelul 7:

Tabel nr. 6

Metrica	**Halstead**	**McCabe**	**LIL**	**C_d**
Halstead	1	0.97	0.91	0.93
McCabe	0.97	1	0.96	0.84
LIL	0.91	0.96	1	0.88
C_d	0.93	0.84	0.88	1

6. Concluzii

Deoarece valorile obtinute pentru indicatorul P (coeficient de corelatie) sunt în intervalul [0.75, 1), toate metricile Halstead, McCabe, metrica dependentelor sunt reprezentative, fiecare din ele putând fi folosita pentru masurarea complexitatii programelor. Altfel spus, oricare metrica din cele trei mentionate anterior se va folosi pentru clasificarea programelor, se vor obtine aceleasi rezultate.

Bibliografie:

[Hals1] Halstead, M.H. - Elements of Software Science Elsevier-North Holland, Amsterdam, 1977

[McCabe1] McCabe, T. J. - A Complexity Measure IEEE Transaction of Software Engineering nr.2(4) 1976, pg. 308-320

[IvArh1] Ion Ivan, M. Macesanu, R. Arhire – Complexity Classes for Software Products, Buletinul roman de informatica, vol4, nr.1, 1985 pg. 63 – 68

[IvArh2] Ion Ivan, R. Arhire – System of Indicators for Estimating the Software Performances, Cybernatics and System Research, R. Trappel Editor, North – Holland Publishing Company, 1982, p.765–771

[Ivan1] Ion Ivan, Adrian Licuriceanu, Lupu Gheorghe – Substituire de metrici software, Revista Româna de Informatica

[Ivan2] Ion Ivan, L. Tovissi, E. Moscovici – Utilizarea analizei entropice in studiul complexitatii programelor, cu aplicatii la normarea activitatii de programare, Buletinul de Informatica, vol.3, nr 4, 1982, pg. 39 – 44

[Ivan3] Ion Ivan, Felix Simion – Gradul de interferenta al metricilor Halstead si McCabe, Revista Româna de Informatica si Automatica, vol.7, nr.3, 1997, p. 23– 28.

[Ivan4] Ion Ivan, Mihai Popescu, Panagiotis Simioros, Felix Simion – Metrici software, Editura Inforec, Bucuresti, 1997