

CODAREA HUFFMAN

1 *Obiectivul lucrării*

Obiectivul principal al lucrării este de prezentare a algoritmului de codare Huffman, precum și de înțelegere a acestuia. În acest scop ne vom folosi de mai multe programe ilustrative.

2 *Introducere teoretică*

În cazul codării simbolurilor individuale algoritmul de codare Shannon-Fano este, în general satisfăcător, conducând la eficiențe ridicate. Pentru $p(u_i) \neq 2^{-l_i}$, algoritmul nu asigură obținerea celei mai reduse lungimi medii a cuvintelor de cod în raport cu alți algoritmi posibili.

Algoritmul propus de D.A. de cod mai mică, în cazul codării simbolurilor individuale. Se consideră, pentru început, cazul codurilor binare, $D=2$. Reamintind că operația de compactare optimă presupune minimizarea lungimii Huffman este optimal, în sensul că nici un alt algoritm de codare nu conduce la un cod de lungime medie a cuvintelor medii a cuvintelor de cod $\bar{L} = \sum p_i l_i$, unde $l_1, \dots, l_i, \dots, l_n$ reprezintă lungimile cuvintelor de cod și sunt întregi care satisfac inegalitatea lui Kraft.

Proprietățile unui cod optimal:

Pentru orice sursă discretă Q -ară, un cod binar ($D=2$) fără prefix, optimal în raport cu minimizarea lungimii medii a cuvintelor de cod, are următoarele proprietăți:

1. Dacă $p(u_j) > p(u_k)$, atunci $l_j \leq l_k$;
2. Ultimelor două simboluri de cea mai mică probabilitate din alfabetul sursei le corespund cuvinte de cod de aceeași lungime;
3. Dacă există două sau mai multe cuvinte de cod de aceeași lungime, atunci două dintre aceste cuvinte diferă numai prin ultimul simbol.

Algoritmul Huffman de codare binară cuprinde următorii pași:

Pas 1: Se ordonează simbolurile sursei primare Q -are în sens descrescător al probabilităților:

$$p(u_1) \geq p(u_2) \geq \dots$$

și se notează sursa primară prin R_0 , adică sursa restrânsă de ordinul 0.

Pas 2: Se grupează ultimele două simboluri având cele mai mici probabilități, u_{k-1} și u_k , într-un simbol artificial r_1 având probabilitatea

$$p(r_1) = p(u_{k-1}) + p(u_k).$$

Pas 3: Se asignează cifra 1 simbolului u_{k-1} și 0 simbolului u_k (sau invers) din grupul r_1 .

Pas 4: Se repetă pașii 1 și 2 pentru noua sursă artificial obținută, numită și sursă restrânsă de ordin întâi R_1 , și, de asemenea, pentru șirul de surse restrânse de ordinul al doilea, R_2 , de ordinul al treilea, R_3, \dots , pana cand se obtine sursa restransa de ordinul $n = Q - 2$, R_n , care furnizează doar două simboluri artificiale.

Pas 5: Cuvântul de cod complet, corespunzător unui simbol al sursei primare, este constituit din secvența literelor codului obținută prin parcurgerea surselor restrânse în sensul opus restrângerii, până la găsirea simbolului original; aceasta echivalează cu parcurgerea unui arbore de la un nod final la rădăcină.

Organigrama algoritmului este prezentată în Fig.1.

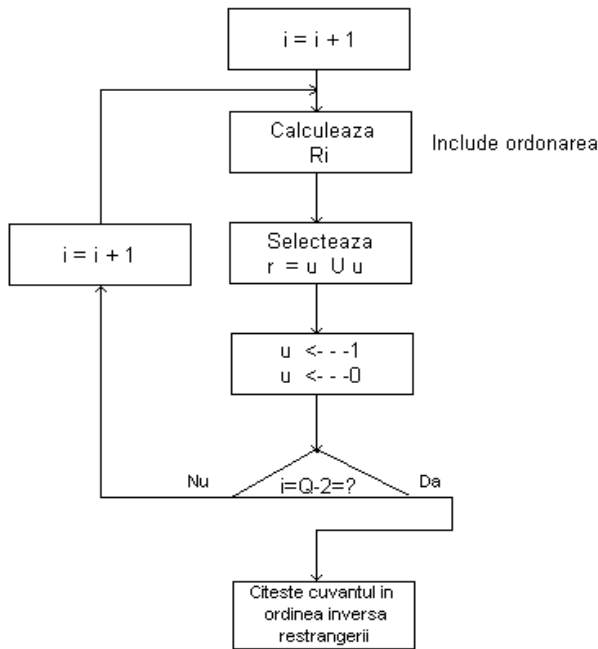


Fig. 1. Organigrama algoritmului Huffman de codare optimala (D=2).

Dacă probabilitățile celor două simboluri reunite în simbolul artificial r_j selectat la fiecare restrângere sunt egale, atunci se poate obține un cod absolut optimal. În caz contrar, se obține un cod optimal cu atât mai apropiat de unul absolut optimal cu cât diferența dintre probabilitățile celor două elemente ale lui r_j este mai mică.

Exemplu 1. Un exemplu de codare Huffman este prezentat sub forma unui graf arbore în Fig 2. Acesta este desenat cu nodurile finale la stânga și rădăcina la dreapta.

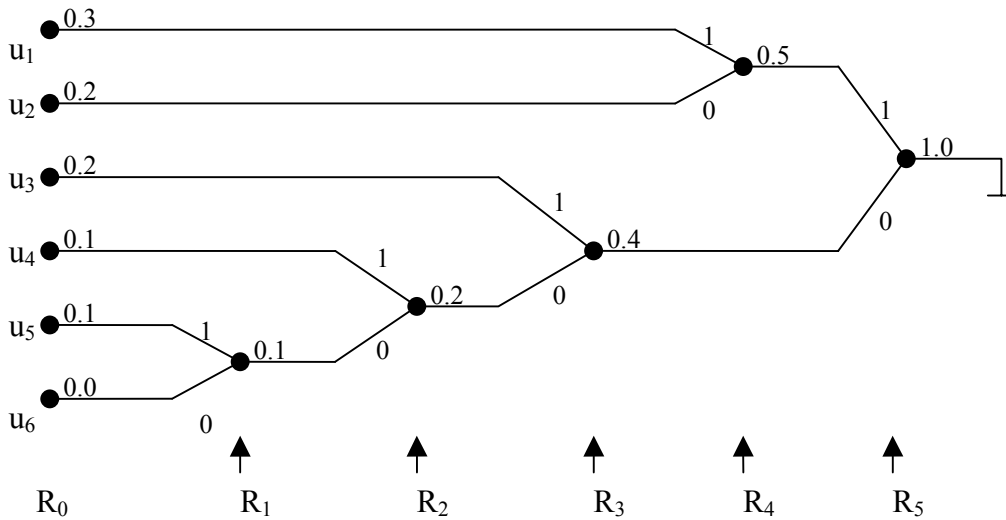


Fig. 2. Graful arbore pentru algoritmul de codare Huffman.

Cuvintele de cod se obțin prin citire de la rădăcină către vârfuri, adică de la stânga la dreapta. De asemenea, se obțin sursele restrânse de ordinul 1 și 2 după cum urmează:

$$C_1 : \begin{cases} c_1 = 11 & c_4 = 001 & R_0 = \{u_1, u_2, u_3, u_4, u_5, u_6\} \\ c_2 = 10 & c_5 = 0001 & R_1 = \{u_1, u_2, u_3, u_4, u_{5,6}\} \\ c_3 = 01 & c_6 = 0000 & R_2 = \{u_1, u_2, u_3, u_{4,5,6}\} \end{cases}$$

Evident, se poate obține un alt cod prin negarea biților fiecărui simbol din cuvintele codului C_1 (se va inversa alocarea bitilor 0 și 1). Se obține codul C_2 :

Dacă se respectă cu strictețe regula ca într-un cuplu de simboluri celui de probabilitate mai mică să i se atribuie 0 și celuilalt 1, se obține codul din Fig. 3. Prin completarea fiecărui simbol din cuvintele lui C_3 se obține:

$$C_2 : \begin{cases} c_1 = 00 & c_4 = 110 \\ c_2 = 01 & c_5 = 1110 \\ c_3 = 10 & c_6 = 1111 \end{cases} \quad C_3 : \begin{cases} c_1 = 11 & c_4 = 010 \\ c_2 = 10 & c_5 = 0111 \\ c_3 = 00 & c_6 = 0110 \end{cases}$$

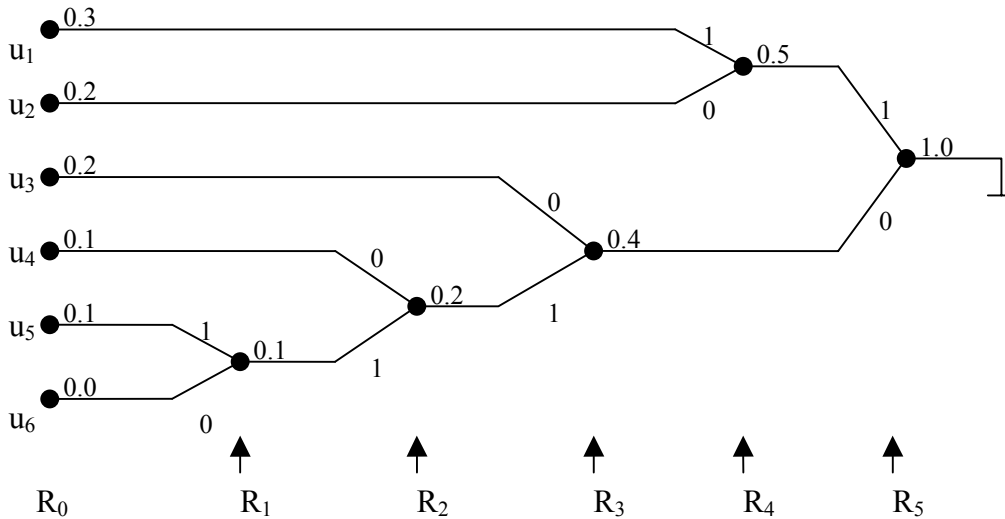


Fig. 3. Varianta de alocare a simbolurilor în algoritmul de codare Huffman.

Analog codului C_2 , se poate obține și pentru codul C_3 complementul său astfel:

$$C_4 : \begin{cases} c_1 = 00 & c_4 = 101 \\ c_2 = 01 & c_5 = 1000 \\ c_3 = 11 & c_6 = 1001 \end{cases}$$

Pentru toate cele 4 coduri C_1, C_2, C_3, C_4 rezultă aceiași parametri caracteristici:

- Entropia sursei $H(U) = 2,3$ bit/simbol,
- Lungimea medie a cuvântului de cod $\bar{L}_1 = 2,4$ simboluri ,
 $\bar{L}_{1\min} = 2,3$ simboluri,
- Eficiența codului $\eta_c = \frac{\bar{L}_{1\min}}{\bar{L}_1} = \frac{2,3}{2,4} = 0,96$,

Parametrii codului construit satisfac prima teoremă a lui Shannon.

$$H(U) < \bar{L}_1 < H(U) + 1 = 3,3.$$

În general, la fiecare sursă restrânsă, alocarea simbolurilor 0 și 1 se poate face în orice ordine, deoarece simbolurile 0 și 1 sunt utilizate pe canal cu aceeași probabilitate. Cum numărul surselor restrânse este Q , iar numărul nodurilor din arborele codării este Q , există în total Q alocări, deci probabilitatea de a construi 2^Q coduri distincte. Toate aceste coduri sunt echivalente în sensul că au aceiași parametri statistici și informaționali.

3 Descrierea evoluției programului

După lansarea în execuție a programului prin "huffman.exe", se afișează un meniu care permite utilizatorului să acceseze codarea Huffman binară (F2), ternară (F3), sau ieșirea din program (Alt-X).

Programul cere introducerea lui n = numărul simbolurilor sursei discrete care urmează a fi codată și a tipului de codare (simbol cu simbol, pe grupe de câte două simboluri sau pe grupe de câte trei simboluri). În cazul codării simbol cu simbol, n poate fi ales între 1 și 8, în cazul codării pe grupe de câte două simboluri, $n=2$ sau 3, iar în cazul codării pe grupe de câte trei simboluri, $n=2$. În celelalte situații, programul va semnala "Invalid data".

Apoi, programul va solicita introducerea valorilor probabilităților simbolurilor sursei. Dacă nu este respectată condiția de normare, programul va semnala "Invalid data".

În continuare, se intră într-un ecran care prezintă arborele de construcție al codului Huffman, pas cu pas, pentru fiecare sursă restrânsă formată, în conformitate cu datele anterior introduse. De asemenea, programul afișează lista cuvintelor de cod obținute, calculează lungimea medie a cuvintelor de cod și eficiența codării.

Codare de compactare

Cu acest program se vizualizează etapele codării Huffman în mai multe variante. Avem opțiunile de codare:

-tasta F2 pentru codare Huffman binar; simbolurile vor fi codate folosind 2 biți (0 și 1);

-tasta F3 pentru codare Huffman ternar; simbolurile vor fi, în aceste caz, codate folosind 3 biți (0, 1 și 2);



Fig4. Alegerea numărului de simboluri ale sursei

Apăsând tasta F2, ne apare meniul din figura 5, în care vom preciza numărul de simboluri ale sursei. Acest număr trebuie să fie între 2 (sursa emite minim 2 simboluri) și 8 (număr ales din motive grafice; pentru un număr mai mare apar dificultăți la afișarea grafului).

Se observă posibilitatea alegerii tipului de codare:

- a) - simbol cu simbol;
- b) - pe grupe de două simboluri;
- c) - pe grupe de trei simboluri;

După alegerea tipului de codare, se vor introduce probabilitățile de apariție a simbolurilor. Vor trebui introduse N-1 probabilități, deoarece pentru ultimul simbol se calculează prin diferență până la 1 probabilitatea acestuia.



Fig5. Fereastra de introducere a probabilităților simbolurilor.

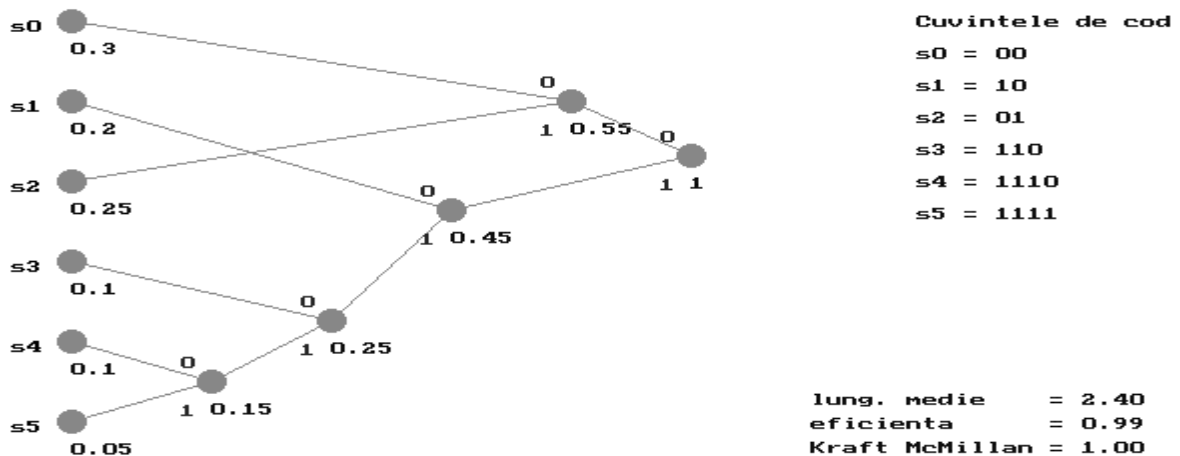
După introducerea probabilităților programul va trece în modul grafic și va afișa etapele de realizare a surselor restrânse de ordinul k , $k \in \{1 \dots N\}$.

Exemplu 2

Spre exemplificare, alegem numărul de simboluri $N=6$ și probabilitățile de apariție:

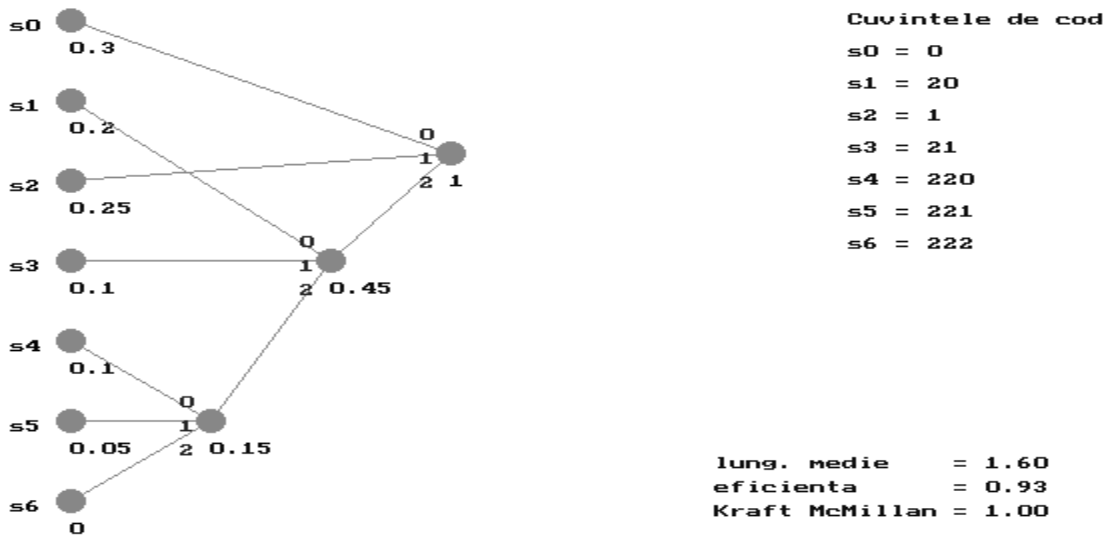
$$\begin{aligned} p(0) &= 0.3 & p(3) &= 0.1 \\ p(1) &= 0.2 & p(4) &= 0.1 \\ p(2) &= 0.25 & p(5) &= 0.05 \end{aligned}$$

După 6 iterații, se afișează graful final, precum și cuvintele de cod optimale rezultate în urma codării. De asemenea se vor afișa valorile parametrilor caracteristici: entropia codului, lungimea medie a cuvintelor de cod, eficiența algoritmului.



Exemplu 2. Graful final al algoritmului Huffman cod binar

Dacă încercăm să realizăm o codare Huffman ternară atunci vom obține rezultatele din Exemplul 2:



Exemplu 2. Graful final al algoritmului Huffman cod ternar.

4 Desfășurarea lucrării

- 4.1 Se intra in optiunea F2, respectiv F3, in cadrul careia se alege n si tipul codarii.
- 4.2 Se introduce setul de probabilitati corespunzator.
- 4.3 Se urmareste constructia codului pentru surse de diferite dimensiuni si statistici, in cazul tuturor celor trei tipuri de codare si se deseneaza graful codarii.
- 4.4 De fiecare data se observa si se noteaza:
 - corespondenta simbol de sursa-cuvant de cod
 - lungimea medie a cuvintelor de cod
 - eficienta codarii
 - valoare lui K rezultata din evaluarea membrului stang al inegalitatii Kraft-McMillan
- 4.5 Se compara parametrii obtinuti in aceste situatii

5 Întrebări

- 5.1 Care sunt proprietatile codurilor obtinute prin algoritmul Huffman?
- 5.2 Care tip de codare (simbol cu simbol, pe grupe de cate 2 simboluri sau pe grupe de cate 3 simboluri) este mai eficient si de ce ?
- 5.3 Codarea Huffman binara/ternara poate duce la obtinerea unui cod absolut optimal (cu $\eta=1$) ? In ce conditii ?
- 5.4 comentați grafurile algoritmului Huffman construite pentru exemplul 2.