

IMPLEMENTAREA TEHNOLOGIILOR INFORMAȚIONALE LA REZOLVAREA UNOR PROBLEME DIN PROBABILITATE

IMPLEMENTATION OF INFORMATION TECHNOLOGIES AT THE SOLVING OF SOME PROBABILITY PROBLEMS

Natalia Neagu, dr., lect. univ.,
UPS „Ion Creangă” din Chișinău

Natalia Neagu, Ph.D., univ.lect.
“Ion Creangă” SPU of Chișinău
ORCID: 0000-0003-3944-3688

CZU: 519.2:004

DOI: 10.46728/c.v2.25-03-2022.p293-298

Rezumat

În acest articol am descris simularea unor problemelor de probabilitate și statistică. Aceste aplicații sunt binevenite în predarea acestui curs, deoarece ne ajută să calculăm probabilitatea teoretică și experimentală într-un timp foarte scurt.

Cuvinte-cheie: tehnologii informaționale, simulare, experiment, probabilitate teoretică și probabilitate experimentală.

Abstract

In this article I describe the simulation of some problems in probability and statistics. These applications are welcome in teaching in this course, as it helps us to calculate theoretical and experimental probability in a very limited time.

Key-words: information technologies, simulation, experiment, theoretical probability and experimental probability.

Instruirea asistată de calculator reprezintă un instrument util și eficient aplicat nu numai în învățământ dar și toate domniile existente. În învățământ există diverse programe destinate învățării printre care se numără și mediul de programare Delphi, în care pot fi realizate programe, astfel încât să prezinte elevului/studentului o cantitate de informație, iar apoi, alternativ, să testeze modul de înțelegere și însușire a informației respective.

Tradițional, cursul de probabilitate și statistică studiază mai mult datele teoretice decât datele experimentale, deoarece ultimele necesită utilaje costisitoare, instrumente, timp, spațiu, etc. De aici și apare confuzia, la elevi/studenti: de ce diferă probabilitatea teoretică și cea experimentală, cu excepția așa numitor *cazuri perfecte* (când coincide probabilitatea teoretică și cea experimentală)? De unde apar aceste erori?

Este bine venit, ca la cursul de probabilitate și statistică, și nu numai la el, să aplicăm unele simulări la calculator care să ne ofere posibilitatea realizării experimentelor, dar și realizarea rapidă a unor calcule necesare.

În această lucrare voi prezenta determinarea probabilității teoretice și probabilității experimentale cu ajutorul simulării unor probleme din probabilitate în mediul de programare Delphi. Însă nu voi prezenta codul programul integral, ci doar cele mai importante secvențe din acesta.

Problema 1. Într-o urnă sunt 100 de bile numerotate de la 1 la 100, fără repetare. Se extrage, aleator o bilă. Să se determine probabilitatea că bila extrasă are imprimat, pe ea, un număr prim.

Rezolvare. Notăm prin X evenimentul extragerii unei bile cu numărul prim (2, 3, 5, 7, ..., 97). Este cunoscut că de la 1 la 100 sunt 25 de numere prime. Pentru a calcula probabilitatea teoretică, vom utiliza formula clasică de calcul a probabilității, adică

$$P(X) = \frac{c_f}{c_t},$$

unde c_f este numărul de cauri favorabile, iar c_t - numărul de cazuri totale. Deci

$$P(X) = \frac{25}{100} = 0,25.$$

Am obținut probabilitatea teoretică de extragere a unei bile cu un număr prim, din 100 bile, numerotate de la 1 la 100, care este $P(X) = 0,25$.

Aceeași condiție a fost simulată în mediul de programare Delphi. Pe forma aplicației, am plasat o imagine care încarcă condiția problemei, dintr-un directoriu curent *img*
`image1.Picture.LoadFromFile('img\1.jpg');`

În componenta *Button 1* (START) este programat experimentul problemei. Cu ajutorul unui ciclu, care se va repeta de n ori, unde n este numărul introdus în *Edit1* (fig. 1)

```
n:=strtoint(edit1.Text);
```

se selectează aleator câte un număr de la 1 la 100 și se verifică dacă numărul selectat este prim. Dacă această condiție este adevărată, atunci bila extrasă se va număra în variabila de contor c ca un caz favorabil

```
c:=0;
```

```
for j:=1 to n do
```

```
begin
```

```
  i:=random(100);
```

```
  if (i=2) or (i=3) or (i=5) or (i=7) or (i=11) or (i=13) or (i=17) or  
  (i=19) or (i=23) or (i=29) or (i=31) or (i=37) or (i=41) or (i=43) or  
  (i=47) or (i=53) or (i=59) or (i=61) or (i=67) or (i=71) or (i=73) or  
  (i=79) or (i=83) or (i=89) or (i=97)
```

```
then
```

```
  c:=c+1;
```

```
end;
```

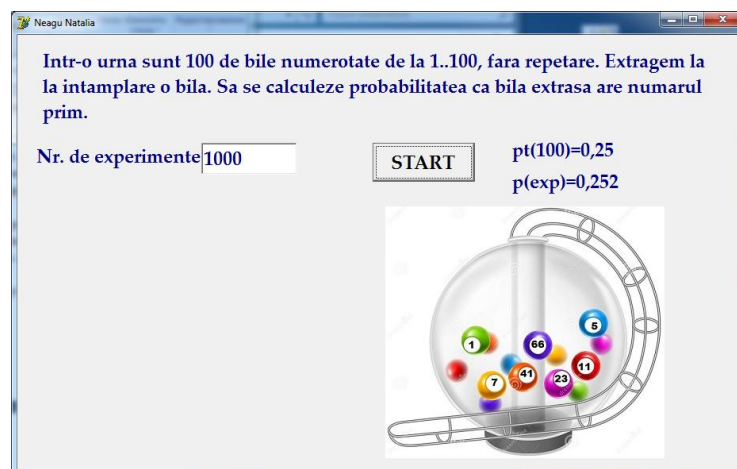


Fig. 1. Simularea experimentului din problema 1

Problema 2. Într-o urnă sunt 26 de bile, pe care sunt indicate câte o literă a alfabetului latin (fără repetare). Se extrag, aleator, fără întoarcere, N bile (N - numărul de litere din cuvântul țintă). Să se determine probabilitatea că literele extrase (în ordinea respectivă) formează cuvântul ARC.

Rezolvare. Notăm prin X evenimentul extragerii a 3 bile succesive, fără întoarcere, care să formeze cuvântului ARC (în ordinea extragerii).

De asemenea notăm următoarele evenimente:

A_1 – la prima extragere să obținem litera A;

A_2 – la a doua extragere să obținem litera R;

A_3 – la a treia extragere să obținem litera C.

Atunci evenimentul X are loc dacă și numai dacă se realizează evenimentele A_1, A_2 și A_3 simultan, ceea ce reprezintă intersecția acestor evenimente

$$X = A_1 \cap A_2 \cap A_3.$$

În conformitate cu formulele de calcul a probabilității, avem

$$P(X) = P(A_1) * P(A_2|A_1) * P(A_3|A_1 \cap A_2),$$

sau

$$P(X) = \frac{1}{26} * \frac{1}{25} * \frac{1}{24} = \frac{1}{15600}.$$

Pentru realizarea experimentului, din problema 2, de asemenea am creat o simulare în mediul de programare Delphi care ne realizează experimentul problemei, dar și ne calculează atât probabilitatea teoretică cât și cea experimentală pentru oricare cuvânt ce satisface condiția menționată mai sus.

În acest caz, pentru realizarea constructivistă a programului, am creat mai multe funcții și proceduri separate care sunt apelate în componenta *Button2* (START).

```

procedure TForm1.Button2Click(Sender: TObject);
var p,pt:extended;
t:integer;
s:string;
begin
randomize;
label1.Caption:="";
label8.Caption:="";
label9.Caption:="";
s:=edit1.Text;
if (unique_str(s)) then
begin
experiment(strtoint(edit2.Text),s,p);
label1.Caption:='Pexp('+edit1.Text+')='+floattostr(p);
nl:=length(edit1.Text);
pt:=1;
for t:=0 to nl-1 do
begin
pt:=pt*(1)/(26-t);

```

```

end;    label8.Caption:='Pt('+edit1.Text+')='+floattostr(pt);
end
else label9.Caption:='Atentie! Literele nu trebuie sa se repete!';
end;

```

De asemenea, în acest buton, după realizarea experimentului și calcularea probabilității experimentale, se calculează și probabilitatea teoretică, după cum rezultă și din codul de mai sus a evenimentului *Button2Click*.

Funcția *unique_str(s)*, este inclusă în condiția instrucțiunii *if*, dacă aceasta este adevărată, atunci se realizează experimentul problemei.

```

function unique_str(str:string):boolean;
var i,j:integer;
begin
for i := 1 to length(str) do
for j := 1 to length(str) do
if(i<>j) then
if str[i]=str[j] then
begin
Result:=false; Exit;
end;
end;
Result:=true;
end;

```

Cu ajutorul acestei funcții se verifică dacă literele introduse în componenta *Edit2* sunt unice (nu se repetă în cuvântul țintă), în caz contrar nu se realizează experimentul, dar se afișează un mesaj de avertizare, care ne atenționează despre repetarea acestor litere în cuvânt, adică încălcăm condiția problemei 2 (fig. 2).



Fig. 2. Cazul nerespectării condiției problemei 2

În procedura *experiment* se realizează experimentul problemei 2, apelându-se funcțiile ajutoare *IndexOfArray*, *In_Array* și *Compare_Arr*.

```

procedure experiment(n:integer;str:string;var prob:extended);
var a,b:array of integer;
z,i,r,c,j:integer;

```

```

s:string;
begin
  z:=length(str);
  c:=0;
  s:="";
  nl:=0;
  setlength(a,z);
  setlength(b,z);
  for j := 1 to n do
    begin
      for I := 0 to z-1 do
        begin
          a[i]:=IndexOfArray(str[i+1],arr);
          repeat
            r:=random(26);
          until not In_Array(r,b);
          b[i]:=r;
          s:=s+arr[r];
        end;
      if (Compare_Arr(a,b)) then c:=c+1;
      s:=s+AnsiString(#13#10);
    end;
  form1.Memo1.Text:=s;
  prob:=c/n;
end;

```

Rezultatul experimentului se afișează în componenta *Memo1*, de pe formă, unde pot fi vizualizate toate rezultatele extragerii, dar și probabilitatea experimentală și cea teoretică, care sunt afișate *Label1* și respectiv *Label8* (fig. 3).

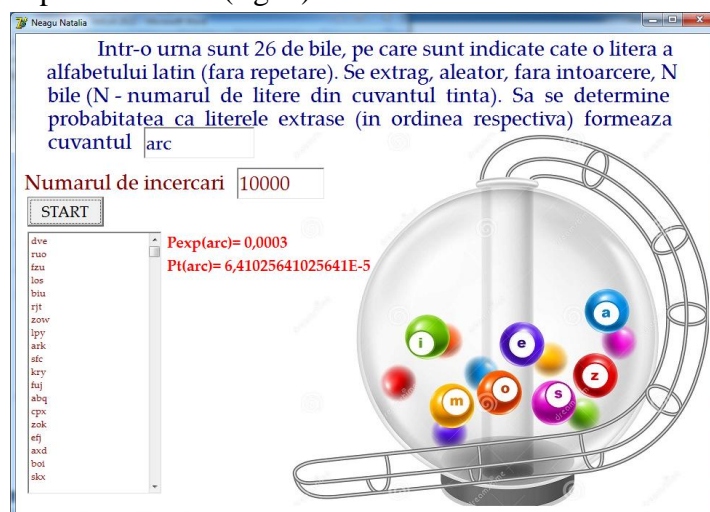


Fig. 3. Simularea experimentului din problema 2

De asemenea avem posibilitatea să modificăm cuvântul țintă. Pentru aceasta, pe formă am introdus o componentă *Edit2*, în care poate fi introdus oricare cuvânt ce satisface condiției problemei 2.

Fie cuvântul țintă MARC, atunci în rezultatul experimentului, vom obține următoarea probabilitatea teoretică și cea experimentală (fig. 4):

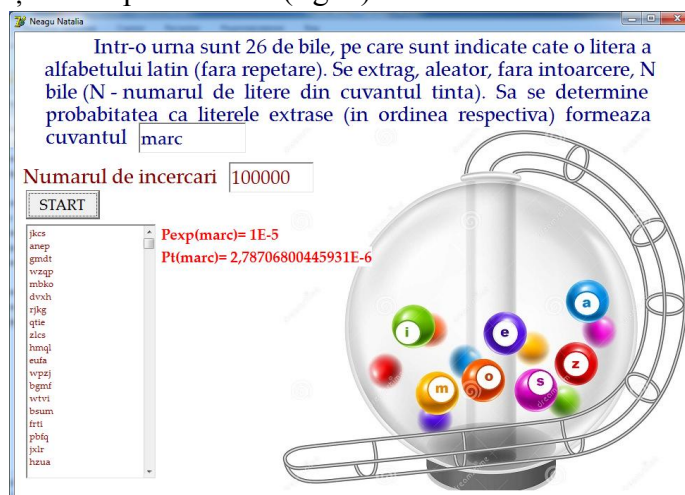


Fig. 4. Rezultatul extragerii cuvântului MARC

Este cunoscut, cu cât lungimea cuvântului țintă este mai mare cu atât probabilitatea extragerii cuvântului dat, va tinde la zero. Din această cauză am selectat cuvintele țintă ARC și MARC, însă programul funcționează pentru oricare lungime posibilă de cuvânt.

Remarcă. Este important să cunoaștem că numărul de încercări contează foarte mult, deoarece cu cât acest număr este mai mare cu atât eroarea de calcul, a probabilității experimentale, este mai mică și valoarea sa este mai aproape de cea teoretică.

Aceste simulări sunt bine venite în cadrul liceelor, dar și în cadrul instituțiilor superioare, unde elevii sau studenții studiază capitolul *Probabilitate și statistică*.

În Republica Moldova, astfel de simulări, nu se realizează, iar ceea ce am realizat este o simulare unică.

În concluzie, menționez că cu cât numărul de experimente este mai mic cu atât probabilitatea experimentală poate varia mai mult față de cea teoretică. O dată nu creșterea numărului de experiențe, eroare de calcul se micșorează, iar rezultatele experimentului devin mai exacte. Simulările unor probleme din probabilitate sunt binevenite în învățământ și pot înlocui unele experimente care necesită utilaje costisitoare, instrumente, timp, etc.

BIBLIOGRAFIE

1. CIUMAC P., CIUMAC V., CIUMAC M. *Teoria probabilităților & elemente de statistică matematică*. Chișinău: Editura "Tehnica" UTM, 2003. 278 p. ISBN 9975-9704-7-8.
2. LUCA-TUDORACHE R. *Probleme de teoria probabilităților*. Iași: Tehnopress, 2006. 100 p. ISBN: 10 973-702-354-4.
3. POPESCU A. *Teoria probabilității. Caiet de seminar*. Petroșani: UNIVERSITAS, 2016. 197 p. ISBN: 978-973-741-512-7.
4. PETREHUS V., POPESCU S. A. *Probabilitate și statistic*. București, 1997, 213 p.