

MODEL MATEMATIC ȘI APLICAȚIE INFORMATICĂ PENTRU LUAREA DECIZIILOR ÎN CONDIȚII DE CERTITUDINE UTILIZÂND METODA MOMENTELOR

Asist.univ.drd. Romana Oancea

Lector univ.dr.ing. Luminița Giurgiu

Academia Forțelor Terestre „Nicolae Bălcescu” Sibiu

Abstract

The paper presents a mathematical model based on inclusion of normal-level risk factors that may influence the tactical missions, for an informatic application of determination of optimal variants. The application was developed in the framework of a research project entitled: „Optimization of capabilities for tactical missions preparation in the field of national defence and security”. This was carried out in the C++Builder environment. The decisional method used in the informatic application uses the method of moments.

În teoria deciziei se folosesc modele deterministe ai căror parametri se calculează prin analiza statistică a datelor care stau la baza selecției. Dacă factorii de risc care pot influența misiunea au valori normale, determinarea soluției se poate face utilizând metoda momentelor;

Ca și etapă premergătoare a determinării soluției optime trebuie făcută o analiză a indicatorilor sintetici ai variației. Acești indicatori caracterizează gradul de variație luând în considerare toți termenii seriei și sunt: abaterea medie liniară, abaterea medie pătratică, dispersia și coeficientul de variație.

- abaterea medie pătratică (abaterea standard), σ – se calculează ca o medie pătratică din abaterile tuturor variantelor seriei de la media lor aritmetică:

$$\sigma = \sqrt{\frac{1}{n} \sum_i (x_i - \bar{X})^2} \quad (1)$$

unde \bar{X} reprezintă media aritmetică

- dispersia, σ^2 - se calculează ca o medie aritmetică simplă sau ponderată a pătratelor abaterilor termenilor față de media lor:

$$\sigma^2 = \frac{1}{n} \sum_i (x_i - \bar{X})^2 \quad (2)$$

- Coeficientul de variație a lui Pearson, v – se obține din raportul abatere medie liniară la nivel mediu sau abatere medie pătratică și medie și poate fi folosit ca test de semnificație a reprezentativității:

$$v = \frac{\sigma_x}{\bar{X}} \% \quad (3)$$

Pentru aplicația de față, dacă coeficientul de variație al unui subcriteriu este sub 2%, nu s-a luat în calcul subcriteriul pentru nici un candidat.

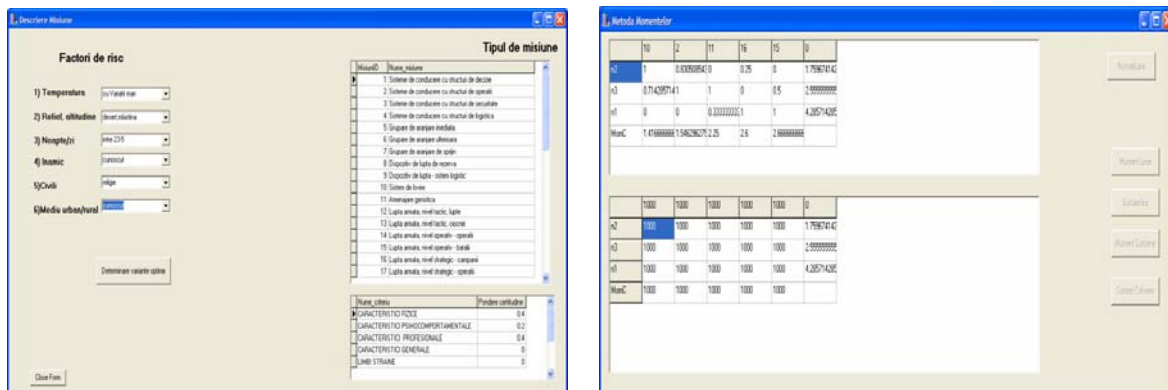


Fig.1 a Descrierea elementelor de risc specifice misiunii
b. Aplicarea, pentru 3 variante, a metodei momentelor

Fiecare factor de risc, în funcție de gradul de incertitudine al său produce o modificare de ponderi pentru subcriteriile care le influențează. În cazul în care factorii de risc care pot influența misiunea au valori normale se poate folosi următorul model matematic înglobat în aplicația informatică proiectă.

Dacă se cunosc toate detaliile misiunii, iar aceasta se desfășoară în condiții de certitudine, subcriteriile sunt echiimportante, se poate obține o soluție optimă utilizând metoda momentelor.

Prelucrarea datelor, obținerea matricei normalizate se face în funcție de tipul criteriilor – maxim sau minim – care au fost luate în calcul și de metoda optimă ce a fost stabilită. În cazul în care criteriile sunt echiimportante se va aplica metoda momentelor.

i) Algoritmul care va rezolva problema decizională este următorul:

- se normalizează matricea consecințelor în funcție de tipul criteriului (minim sau maxim);
- pentru fiecare linie se calculează momentul linie, cu relația (4), după care se ordonează liniile matricei corespunzătoare normalizate în ordine crescătoare după valorile momentelor linie;

$$M_i^l = \frac{\sum_{j=1}^n j \cdot r_{ij}}{\sum_{j=1}^n r_{ij}}, (\forall) i = \overline{1, m} \quad (4)$$

- pentru fiecare coloană a noii matrice se calculează momentul coloană:

$$M_j^c = \frac{\sum_{i=1}^m i \cdot r_{ij}}{\sum_{i=1}^m r_{ij}}, (\forall) j = \overline{1, n} \quad (5)$$

- se ordonează coloanele matricei în ordine crescătoare a valorilor momentelor coloană;
- se reia algoritmul până când nu mai sunt posibile noi ordonări. Ultima ordonare a liniilor reprezintă clasamentul optim al variantelor decizionale.

ii) Implementarea algoritmului

Se interoghează tabela Ofițeri. Pentru toți ofițerii care au testele introduse se construiește o matrice ce are pe prima coloană numele candidaților, pe prima linie identificatorul subcriteriului, iar ca și valori punctajele obținute de aceștia.

```

Table_Ofiteri->Active=true;
Table_Subcrit->Active=true;
StG->RowCount=Table_Ofiteri->RecordCount+2; //definesc liniile
matricei
StG->ColCount=1;//inițial matricea are o singură coloană
//completez liniile coloanei 0 cu numele candidaților
Table_Ofiteri->First(); //mă poziționez pe prima înregistrare în tabela
Ofițeri
i=1;
while(!Table_Ofiteri->Eof)
{
StG->Cells[0][i]=Table_Ofiteri->FieldByName("Nume")->AsString;
Table_Ofiteri->Next();
i++;
}
//completez coloanele(prima linie) cu identificatorul pentru subcriteriu
Table_Temp->Active=true;

```

```

Table_Subcrit->First();
j=1;
while(!Table_Subcrit->Eof)
{
id_subcrit=Table_Subcrit->FieldByName("SubcriteriuID")->AsInteger;
subcriteriu_tip=Table_Subcrit->FieldByName("Tip_sub")->AsString;
if(Table_Temp->Locate("SubcriteriuID",Variant(id_subcrit),Opts))
{
//iau in considerare numai subcriteriile cu o variație mai mare decât 0.02
if(Table_Temp->FieldByName("Variatia")->AsFloat >= 0.02)
{
//am gasit subcriteriul in tabela TEMP
StG->ColCount=StG->ColCount+1;
StG->Cells[j][0]=Table_Subcrit->FieldByName("SubcritID")-
AsString;
id_ofiter=Table_Temp->FieldByName("OfiterID")->AsInteger;
id_criteriu=Table_Subcrit->FieldByName("CriteriuID")->AsInteger;
j++;}
}
Table_Subcrit->Next();
}
StG->ColCount=StG->ColCount+1; //adaug o coloană pentru
calculul momentului linie

```

```

/completez valorile subcriteriilor pentru fiecare ofiteri
//aici trebuie să pun valorile din Temp
Table_Ofiteri->First();
i=1;
while(!Table_Ofiteri->Eof)
{
id_ofiter=Table_Ofiteri->FieldByName("OfiterID")->AsInteger;
loc_values[0]=id_ofiter;
for(j=1;j<StG->ColCount;j++)
{
id_subcrit=StrToInt(StG->Cells[j][0]);
loc_values[1]=id_subcrit;
if(Table_Temp->Locate("OfiterID; SubcriteriuID",
VarArrayOf(loc_values,1), Opts)
StG->Cells[j][i]=Table_Temp->FieldByName("Punctaj")->AsFloat;
else StG->Cells[j][i]=0;
}
Table_Ofiteri->Next();
i++; }

```

Normalizarea se face în funcție de tipul subcriteriului. Se determină mai întâi minimul sau maximum pe fiecare subcriteriu.

Dacă subcriteriul este de minim, atunci valoarea normalizată se calculează cu relația:

$$r_{ij} = \frac{a_{j\max} - a_{ij}}{a_{j\max} - a_{j\min}} \quad (6)$$

Dacă subcriteriul este de maxim, noua valoare se determină cu relația:

$$r_{ij} = \frac{a_{ij} - a_{j\min}}{a_{j\max} - a_{j\min}} \quad (7)$$

```
//normalizarea
for(i=1;i<StG->ColCount-1;i++)
{
sub_crit=(StG->Cells[i][0].c_str());
Query1->Close();
Query1->SQL->Clear();
Query1->SQL->Add("SELECT * FROM Subcrit WHERE
SubcriteriuID="+sub_crit);
Query1->Open();
tip=Query1->FieldName("Tip_sub")->AsString;
//caut maximumul si min in coloana
a_min=StrToFloat(StG->Cells[i][1]);
a_max=StrToFloat(StG->Cells[i][1]);
for(j=1;j<StG->RowCount-1;j++)
{
if(a_min>StG->Cells[i][j]) a_min=StrToFloat(StG->Cells[i][j]);
if(a_max<StG->Cells[i][j]) a_max=StrToFloat(StG->Cells[i][j]);
}
if(tip=="max")
//normalizez cu MAX
for(j=1;j<StG->RowCount-1;j++)
{
a=StrToFloat(StG->Cells[i][j]);
//ShowMessage(a);
a=double(double(a-a_min)/double(a_max-a_min));
StG->Cells[i][j]=FloatToStr(a);
}
else
for(j=1;j<StG->RowCount-1;j++)
{
a=StrToFloat(StG->Cells[i][j]);
a=double(double(a_max-a)/double(a_max-a_min));
StG->Cells[i][j]=FloatToStr(a);
}
}
}
```

```

    }
}
Calculez momentul linie:
for (i=1;i<nr_lin-1;i++)
{
    suma=0;
    suma_numar=0;
    for (j=1;j<nr_col-1;j++)
    {
        suma_numar=suma_numar+j*double(StrToFloat(StG->Cells[j][i]));
        suma=suma+StrToFloat(StG->Cells[j][i]);
    }
    mom_l=double(double(suma_numar)/double(suma));
    StG->Cells[j][i]=FloatToStr(mom_l);
}

```

După ordonarea crescătoare în funcție de momentul linie, calculez momentul coloană:

```

for (i=1;i<nr_col-1;i++)
{
    suma=0;
    suma_numar=0;
    for (j=1;j<nr_lin-1;j++)
    {
        suma_numar=suma_numar+j*(StrToFloat(StG1->Cells[i][j]));
        suma=suma+StrToFloat(StG1->Cells[i][j]);
    }
    mom_c=double(double(suma_numar)/double(suma));
    StG1->Cells[i][j]=FloatToStr(mom_c);
}

```

Se sortează crescător în funcție de momentul coloană. Algoritmul, după cum am mai spus, se repetă până când nu se mai produce nici o schimbare între liniile și coloanele matricei.

Bibliografie

- [1] Bell, D., Raiffa, H., Tversky, A., *Decision making: descriptive, normative and prescriptive interactions*, Cambridge University Press, Cambridge, 1988.
- [2] Miclea, M., *Psihologie cognitivă-modele teoretico-experimentale*, Editura Polirom, Iași, 2003;
- [3] Paul Weirich, *Realistic Decision Theory: Rules for Nonideal Agents in Nonideal Circumstances*, Oxford University Press, 2004.