

Laborator 2: Modelarea unei probleme de decizie

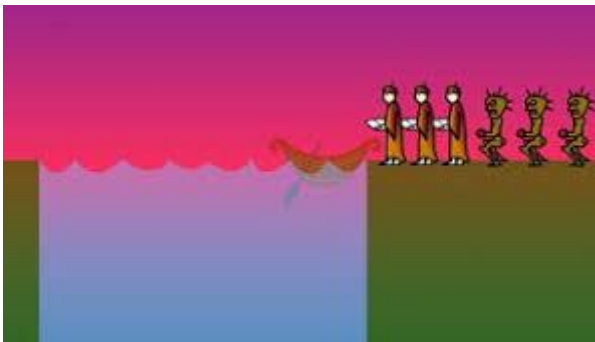
Problema: Fie m misionari și c canibali ($m \geq c$) pe malul stâng al unui râu. Pe același mal se află o barcă de capacitate b (în barcă încap cel mult b persoane). Găsiți o secvență de transferuri de persoane de pe un mal pe celălalt, folosind barca, astfel încât toate persoanele să ajungă pe malul drept al râului. Barca nu își poate schimba poziția (malul) decât transferând cel puțin o persoană pe noul mal. În nici un moment, pe nici unul din maluri, nu pot exista mai mulți canibali decât misionari.

Etape de rezolvare:

1. Alegeți o reprezentare a unei stări a problemei. Reprezentarea trebuie să fie suficient de explicită pentru a conține toate informațiile necesare pentru continuarea găsirii unei soluții dar trebuie să fie și suficient de formalizată pentru a fi ușor de prelucrat/memorat.
2. (0.2) Identificați stările speciale (inițială și finală) și implementați funcția de inițializare (primește ca parametri instanța problemei, întoarce starea inițială) și funcția booleană care verifică dacă o stare primită ca parametru este finală.
3. (0.2) Implementați tranziția ca o funcție care primește ca parametri o stare și parametrii tranziției și întoarce starea rezultată în urma aplicării tranziției. Validarea tranziției se face într-o funcție booleană separată, cu aceeași parametri.
4. (0.2) Implementați strategia random, cu optimizările:
 - a. Nu alege o stare vizitată anterior.
 - b. Contorizează numărul de tranziții făcute efectiv, dacă după 100 de tranziții nu a ajuns la starea finală revine la starea inițială și resetează contorul.
5. (0.2) Implementați strategia backtracking.
6. (0.2) Implementați strategia IDDFS (Iterative Deepening Depth First Search).

Sugestii de rezolvare (Discuție începută la laborator)

Alegerea unei reprezentări pentru o stare



Varianta 1:

Explicită dar greu de prelucrat.

Varianta 2: Listă cu pozițiile oamenilor (D,D,D,D,D,D) pentru 3 misionari și 3 canibali, toți pe malul drept. Reprezentare insuficient de explicită, nu știm unde e barca.

Varianta 3:

Instance: nc - number of cannibals, nm - number of missionaries, b - capacity of the boat

State: $(b, nm_1, nc_1, bp, nm_2, nc_2)$, $b \in [1, 2]$, position of the boat, $nm_1 + nm_2 = nm$, $nc_1 + nc_2 = nc$

Funcția de inițializare

Initial state: $(b, nm, nc, 1, 0, 0)$

State Initialize (int nm, int nc, int b)

```
{  
State s = new State(b,nm,nc,1,0,0);  
Return s;}
```

Verificarea stării finale

Final state: (b,0,0,2,nm,nc)

Boolean isFinal(State s)

```
{  
If ((s[1]==0)&&(s[2]==0)&&(s[3]==2)) return true;  
else return false;}
```

Discuțați spațiul problemei (dimensiune, complexitate)

Transitions: (b,nm₁,nc₁,bp,nm₂,nc₂)->(b,nm₁+mm,nc₁+ cm,3-bp,nm₂-mm,nc₂- cm)

State Transition(State s, int mm, int cm)

```
{  
State ns=new State(s[0],s[1]+mm,s[2]+cm,3-s[4],s[5]-mm,s[6]-cm);  
return ns;}
```

Valid transitions:

1. If nm₁>0, nm₁≥nc₁
2. If nm₂>0, nm₂≥nc₂
3. If mm≥0 and cm≥0 and mm + cm >0 then bp=1
4. If mm≤0 and cm≤0 and mm + cm <0 then bp=2
5. |mm+cm|≤b

Boolean Validation(State s, int mm, int cm)

```
{  
State ns=new State(s[0],s[1]+mm,s[2]+cm,3-s[4],s[5]-mm,s[6]-cm);  
if((s[1]>0)&&(s[1]<s[2])) return false;  
else  
if((s[4]>0)&&(s[4]<s[5])) return false;  
else  
if(mm+cm>s[0]) return false;  
else  
if((mm+cm>0)&&(mm>=0)&&(cm>=0)&&(s[3]==2)) return false;  
else if((mm+cm<0)&&(mm<=0)&&(cm<=0)&&(s[3]==1)) return false;  
return true;}
```

Strategia

Void strategy(State)

```
{  
While (not (isFinal(State))  
{  
    Choose mm, cm from possible transitions;  
    if(Validation(State, mm, cm)) State=Transition(State, mm, cm);  
}}
```

Aici sunt apelate cele trei strategii:
random, BKT și IDDFS.