

## Documentatie – Problema de cautare Lupi, capre si verze

Euristica :

Euristica banala:

```
if tip_euristica=="euristica banala":
    if testeaza_scop(infoNod)==False:
        return 1
    return 0
```

Pentru orice stare care nu este finala, se returneaza 1, altfel, 0.

Pentru cele doua euristici admisibile, se ia in calcul doar ce a ramas pe malul de est, indiferent daca taranul este pe malul de vest sau de est (daca este pe malul de vest, oricum va avea un drum in plus pentru a se putea intoarce)

Euristica admisibila 1 :

```
elif tip_euristica == "euristica admisibila 1": #daca sunt mai mult de 3 tipuri de
elemente
    h = 0 #pe malul de est si starile finale
    trebuie sa
    nr = 0 #aiba un nr nenul din fiecare element pe
malul de vest,
    if(infoNod[1][0]!=0): nr+=1 #barca va face cel putin 2 drumuri pt ca
nu se pot
    if(infoNod[1][1]!=0): nr+=1 #transporta decat maxim 2 tipuri de
elemente odata
    if(infoNod[1][2]!=0): nr+=1
    if(nr == 3 and verze_FINAL!=0 and capre_FINAL!=0 and lupi_FINAL!=0): h = 2
    elif(nr > 0): h = 1
    return h
```

Daca pe malul de est au ramas un nr nenul de elemente, se returneaza :

- 2 daca au ramas 3 tipuri de elemente (si verze, si capre, si lupi), deoarece la un drum nu se pot transporta decat maxim doua tipuri de elemente diferite
  - 1 daca au ramas cel putin unul si maxim doua tipuri de elemente diferite
  - 0 daca nu a mai ramas nimic (stare finala)
- ⇒ Mereu subestimeaza (considera ca pentru doua tipuri de elemente, indiferent cate au ramas si daca au loc in barca, este suficient un drum)

Euristica admisibila 2 :

```
elif tip_euristica == "euristica admisibila 2": #nr total de elemente ramase pe malul
de est
    total = K1 + K2 #impartit la nr de locuri disponibile
in barca
    vcl = infoNod[1][0]+infoNod[1][1]+infoNod[1][2]
    h = math.ceil(vcl/total)
    return h
```

- Nr total de elemente ramase pe malul de est (care trebuie transportate pe malul de vest)/nr de locuri disponibile in barca. Nu se poate mica numarul, deoarece stările sunt considerate deja “simplificate” – adica elementele care urmau sa fie eventual mancate au fost mancate deja.
- De asemenea, toate elementele trebuie transportate pe malul de vest deoarece conditia de stare finala presupune sa nu mai ramana nimic pe malul initial
- ⇒ Subestimeaza deoarece considera ca toate elementele ramase pot fi transportate impreuna la fiecare drum, indiferent daca sunt tipuri diferite in acelasi compartiment pe barca, si ia in considerare barca la capacitate maxima => numar minim de drumuri

Euristica neadmisibila :

```
elif tip_euristica == "euristica neadmisibila":    #fiecare element este transportat separat
    h = infoNod[1][0]+infoNod[1][1]+infoNod[1][2]
    return h
```

- Se considera cate un drum separat pentru fiecare element => supraestimeaza  
Ex :  
print(calculeaza\_h([[1],[16,3,2],[2,4,1,0,0]], "euristica neadmisibila")) => afiseaza 21  
Pentru K1 = 3 si K2 = 4 ar fi suficiente 4 drumuri (1. 3 capre+2 lupi, 2. 7 verze, 3. 7 verze, 4. 2 verze)

### Eventuale optimizari/validari :

- Dupa citirea din fisier (verificare daca datele introduse nu sunt negative, daca starea initiala este deja stare finala – se intampla doar daca pe malul de est sunt 0 verze, 0 capre, 0 lupi, iar Starea finala are cerinta tot 0 verze, 0 capre, 0 lupi -, sau daca problema nu are solutii – daca in starea initiala sunt mai putine elemente decat se cer la final)

```
- #Verificare:

#0. Nu sunt numere negative printre datele de intrare
if(capre_init<0 or verze_init<0 or lupi_init<0 or verze_FINAL<0 or\
    capre_FINAL<0 or lupi_FINAL<0 or K1<0 or K2<0 or M<0):
    print("Input invalid")
    exit(True)

#1. Starea initiala este si finala
if(verze_FINAL+capre_FINAL+lupi_FINAL+verze_init+capre_init+lupi_init)==0:
    print("Starea initiala este deja si stare finala")
    exit(True)

#2. Nu se poate ajunge din starea initiala intr-o stare finala
if(verze_init< verze_FINAL or capre_init<capre_FINAL or lupi_init<lupi_FINAL):
    print("Problema nu are solutii")
    exit(True)
```

- De asemenea, reprezentarea unei stari este simplificata -> nu se considera stari intermediare cu barca pe drum, iar in lista de succesori sunt adaugate stari doar odata ce au fost “simplificate” (adica toate animalele care trebuiau sa manance au mancat)