

# Documentatie proiect KR

## 1. Muntele Olimp

Am folosit o functie euristica bazata pe distanta intre nodurile scop si restul nodurilor din graf, practic distanta minima de la oricare nod pana la un nod scop. Aceasta euristica este eficienta, deoarece este admisibila si reprezinta distanta efectiva in graf intre noduri. Ca si implementare, folosesc un dictionar cu fiecare nod si o parcurgere BFS din nodurile scop, actualizand aferent dictionarul in functie de ce noduri parcurg.

Ponderile intre noduri sunt 1, 2, respectiv 3, in functie de distanta dintre ele. Am ales aceste dimensiuni cu gandul ca intre patrate si in patratul cel mic ponderile sunt de o unitate, in patratul mijlociu sunt de 2 unitati, iar in patratul cel mare din exterior sunt de 3 unitati. Se respecta proprietatea din cerinta:  $\text{cost}(1, 2) > \text{cost}(4, 5) > \text{cost}(7, 8) = 3 > 2 > 1$ .

Algoritmul A\* afiseaza lista open dupa numarul de pasi dorit, iar algoritmul IDA\*, in lipsa listei close din implementare, afiseaza drumul pana la un nod scop daca acesta a fost gasit, sau mesajul "Nu s-a gasit un nod scop" in cazul in care dupa numarul de pasi nu s-a gasit un nod scop.

## 2. Nine men's morris

Euristica este reprezentata de scorul unei stari de joc. Am salvat in clasa Game toate configuratiile de pozitii ale morilor posibile in acest joc, iar pe baza acestor configurari, pot sa calculez euristica nodului (starii) astfel:

- `scor_secventa()` calculeaza scorul unei mori date ca parametru:  $10^{\text{nr simboluri in secventa} - 1}$  daca jucatorul opus nu e in secventa, 0 altfel
- `estimare()` foloseste functia `scor_secventa()` pentru a calcula punctajul total al starii, adica suma tuturor configurarilor pozitiiilor atat ale jucatorului MAX, cat si ale jucatorului MIN

Astfel, mutarile sunt determinate in functie de acest scor de estimare si de jucatorul curent care urmeaza sa mute

Algoritmii de MinMax si AlphaBeta determina starea jocului dupa adancimea data ca si parametru, fiecare jucator facand mutarea optima.

## 3. Bayesian Approach

Am ales sa calculez probabilistic urmatoarea mutare, calculand scorul fiecărei posibile mutari din pozitia initiala. Pentru fiecare simbol al lui MAX din moara curenta adaug 0.3 la probabilitatea totala (sau 0.1 daca nu am nici un simbol), iar daca pe moara curenta exista un simbol al lui MIN, probabilitatea va fi 0