

# Problema blocurilor

Blocuri de intervale  $[a,b]$

O mutare se poate efectua doar dacă intervalele celor 2 blocuri (cel mutat și baza) se intersectează.

Nu este permisă mutarea unui bloc pe un alt bloc cu același interval).

Scopul este ca fiecare bloc să se afle în intervalul blocului de sub el.

Costul unei mutări este  $b - a$ .

Librării utilizate:

```
pip install stopit
```

 (Instalare pachet stopit.)

Comanda de rulare:

```
python [folder_input] [folder_output] [numar_solutii_cautate]  
[timer_timeout]
```

## Euristici

Euristicile folosite:

- euristica banala
- euristica admisibila 1
- euristica admisibila 2
- euristica admisibila 3 (euristica șef)
- euristica neadmisibila

Euristica banala:

Așa cum îi spune numele, euristica banală nu calculează nimic special.

$h(\text{nod}) = 0$ , dacă nodul este scop, altfel este 1.

Euristica admisibila 1:

Determinăm pentru fiecare stivă dacă este necesară o mutare.

Parcurgem fiecare stivă de jos în sus. Dacă intervalul de sus nu este inclus în interiorul intervalului curent, este necesară o mutare.

Părăsim stiva dacă am găsit un interval ce trebuie mutat sau dacă am parcurs întreaga stivă.

[4,5]		[4,6]
[3,6]		[3,4]
[1,5]	[2,3]	[3,7]
1	2	3

De exemplu, în figura de mai sus există două intervale care se află nepotrivit în stivă, anume [3,6] din stiva 1 și [4,6] din stiva 3. Astfel, numărul minim de mutări necesare pentru a ajunge la o stare scop este  $h(n) = 2$ .

#### Euristica admisibilă 2:

Are același concept ca euristica admisibilă 1. În plus, dacă intervalul de mai sus nu este inclus în intervalul curent, toate intervalele de deasupra intervalului curent trebuie mutate.

Din exemplul anterior, conform euristicii, reies 3 mutări necesare pentru a ajunge în stare scop. De pe stiva 1 trebuie mutate intervalele de la [3,6] în sus (adică și [4,5], iar de pe stiva 3 trebuie mutat intervalul [4,6]. Astfel, numărul minim de mutări necesare pentru a ajunge la o stare scop este  $h(n) = 3$ .

#### Euristica șef:

Folosește același concept ca euristica admisibilă 2. În plus, ne folosim de faptul ca fiecare mutare de interval are un cost, acela fiind dimensiunea intervalului – 1. (pentru [a,b] costul mutării este  $b - a$ )

Pentru același exemplu, costurile intervalelor ce trebuie mutate ([3,6], [4,5], [4,6]) este  $3 + 1 + 2 = 6$ .

Această euristică este admisibilă, deoarece nu poate fi mai mare decât costul real al drumului către nodul scop, deoarece știm că acesta este costul minim pentru a ajunge la o stare scop.

#### Euristica neadmisibilă:

O euristică este neadmisibilă atunci când supraestimează distanța de la nodul curent la nodul scop. O metodă de a face euristică șef să fie admisibilă, este să înmulțim costul fiecărui interval ce trebuie mutat cu valoarea maximă dintre intervalele ce trebuie mutate.

[4,5]		
[3,6]		
[1,5]	[2,7]	[3,7]
1	2	3

De exemplu, intervalele ce trebuie mutate sunt [3,6] și [4,5]. Costul maxim dintre intervale este  $6 - 3 = 3$ . Numărul minim de mutări este 2. Astfel, estimarea este  $2 * 3 = 6$  ceea ce face ca euristică să fie neadmisibilă, deoarece costul real al acestei stări de a ajunge la nodul scop este de  $3 + 1 = 4$ .

		[3,6]	[4,5]
[1,5]	[2,7]	[3,7]	
1	2	3	

## Algoritmi

Algoritm/Caracteristici	BFS	DFS	DFI	UCS
Lungimea drumului	9	10	9	9
Numărul maxim de noduri existente în graf	7	10	9	6
Numărul de noduri generate	41	13	12	36
Avantaje/Dezavantaje	Cel mai scurt drum(înălțime)/ Ocupa cea mai multa memorie.	/Rămânem fără recurență la soluții în dreapta arborelui.	O combinatie eficienta intre BFS si DFS/Timpul de executie	Oferă drumul de cost minim/Mult mai lent decât A*.

	A*	A* optim	IDA*	Greedy
Lungimea drumului	Timeout	13	Timeout	Timeout
Numar de noduri generate		59189		
Numar de noduri in memorie		34386		
Avantaje/Dezavantaje	Cel mai bun algoritm pentru a genera NSOL de drum minim./Nu poate fi la fel de performant ca A* pe o singura solutie	Transmite o singura solutie foarte rapida de cost minim/Doar o solutie generata	O varianta la fel de buna ca A*, ocupa mai putin spatiu in memorie/Dureaza mai mult timp pentru a gasi solutia.	Ofera solutia repede/Nu este cea mai scurta intotdeauna