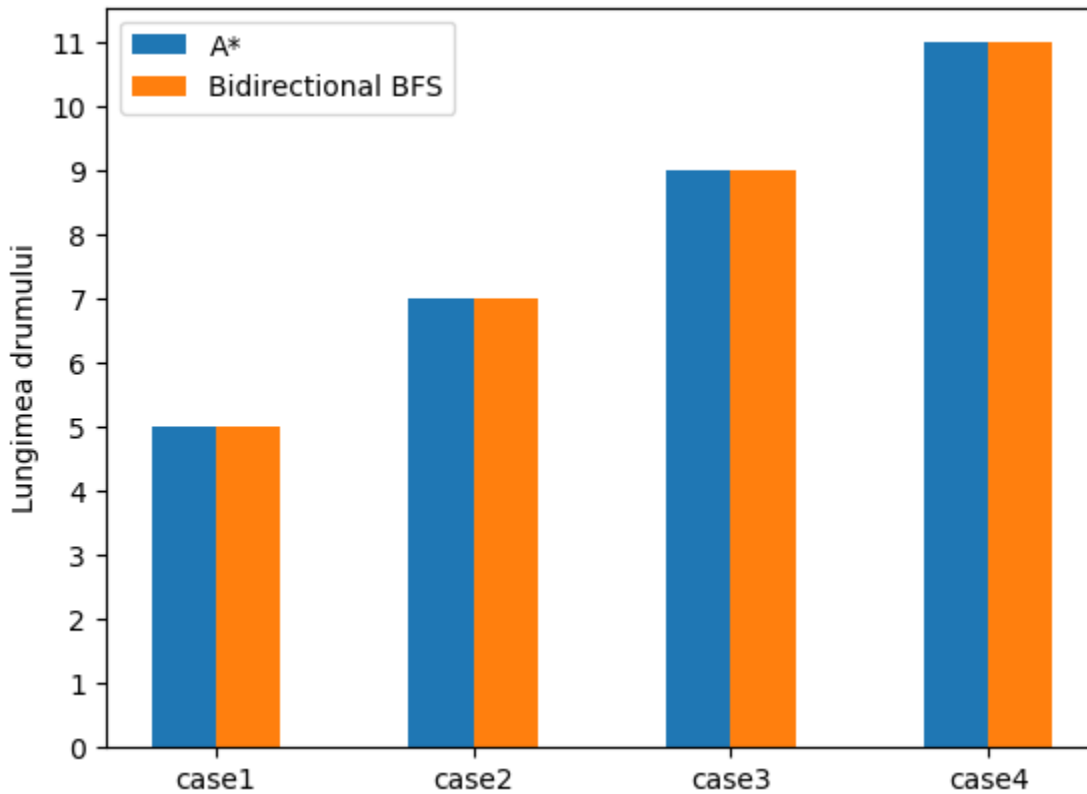


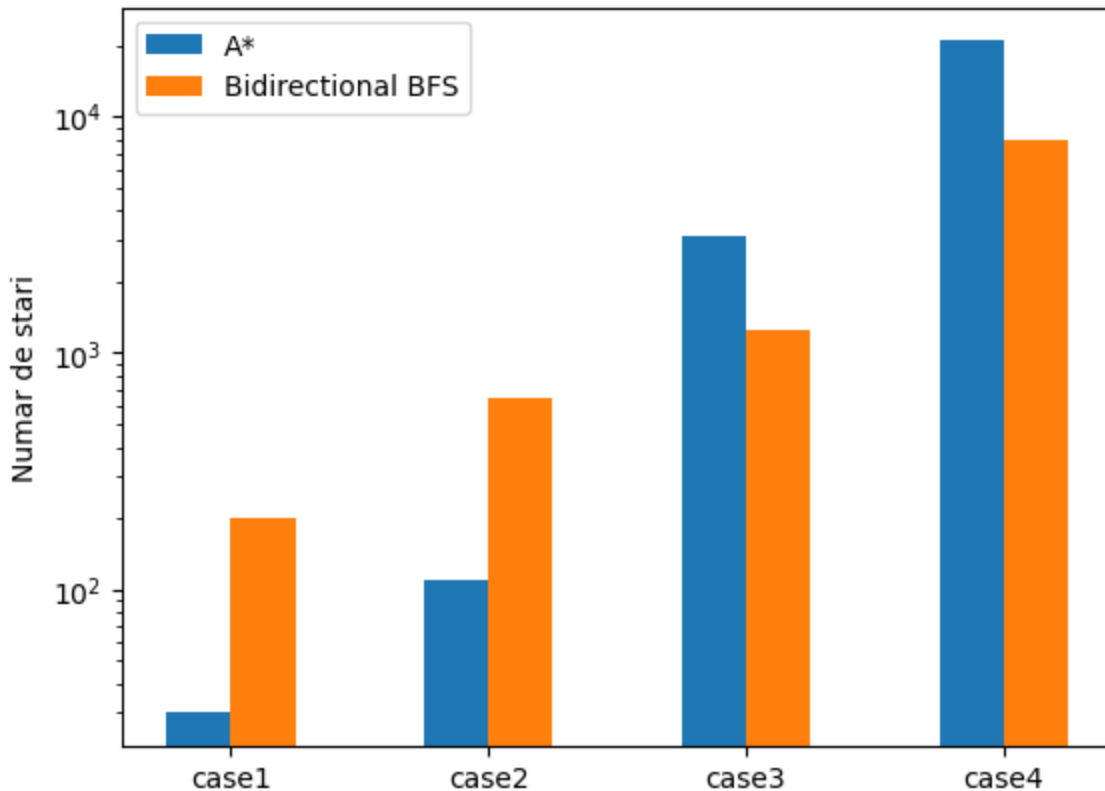
Nume: Vîrlan Mihnea-Alexandru  
Grupa: 341C4

## Tema 1 - Pocket Cube

### Cerința 1 - A\* și BFS Bidirecțional



În bar chart-ul de mai sus, se poate observa că A\* și BFS Bidirecțional estimează același drum minim de la starea inițială până la soluție. Dacă la BFS-ul Bidirecțional am fi ajuns garantat la o soluție optimă, în cazul A\* am ajuns la soluția optimă datorită euristicii admisibile implementate. Am recurs la o euristică admisibilă descrisă în acest articol al lui Richard E. Korf [1] în care euristica manhattan este dată de numărul minim de mutări necesare pentru fiecare cubuleț/sticker pentru a ajunge la poziția și orientarea corectă a culorilor. În helpers.py se găsesc detalii suplimentare despre implementarea euristicii.



În graficul de mai sus avem evoluția numărului de stări pentru cei doi algoritmi. Cum era de așteptat, evoluția acestui număr este exponențială. Se poate observa în cazurile de test case1 și case2 că BFS-ul bidirecțional generează mai multe stări decât A\*, acest lucru fiind datorat factorului de ramificare  $b^*$ , care evaluează calitatea euristicii (conform [2]).

Din [2] folosim următoarea formulă pentru a determina  $b^*$

$$N + 1 = 1 + b^* + (b^*)^2 + \dots + (b^*)^d.$$

Unde  $N$  este numărul de stări generate, iar  $d$  este adâncimea soluției. De reținut este că graficul de mai sus, numărul de stări cuprinde și starea inițială.

Pentru case1, avem  $N+1=30$  și  $d=5$ . Vom avea  $b^*=1,657$ .

Pentru case2, avem  $N+1=109$  și  $d=7$ . Vom avea  $b^*=1,731$

În notebook-ul pentru Cerința 1 am introdus o funcție care rezolvă ecuația folosind metoda Newton-Raphson.

Valorile pentru  $b^*$  indică faptul că, pentru euristica ce calculează numărul minim de mutări făcute de fiecare cubuleț pentru a ajunge în poziția corectă (detalii în helpers.py), algoritmul A\* tinde să găsească mai multe situații când numai o singură stare vecină de la nivelul următor va fi pusă în coada de priorități (mulțimea OPEN). Asta face ca aceste căutări să se termine mai repede. Prin urmare, pentru cazuri precum case1, sau case2, deci de dificultate cel mult medie, A\* e de preferat.

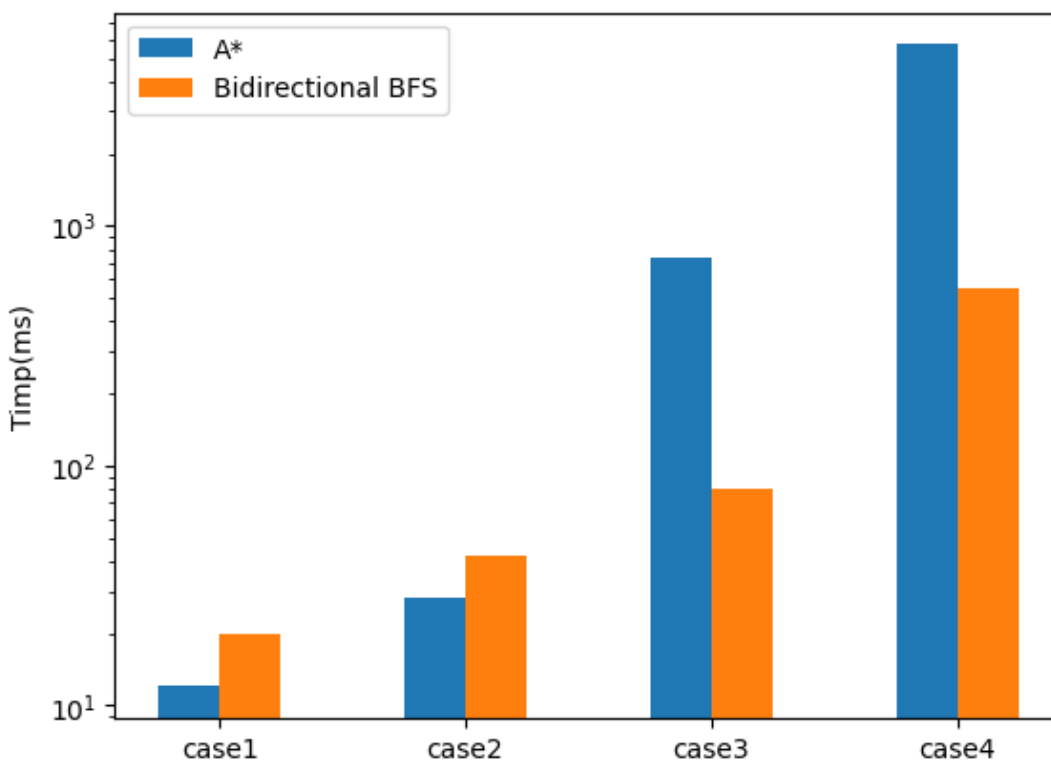
Pentru ultimele două cazuri, valorile pentru  $b^*$  sunt  
Pentru case3,  $N+1=3145$  și  $d=9$ . Vom avea  $b^*=2,296$   
Pentru case4,  $N+1=20900$  și  $d=11$ . Vom avea  $b^*=2,3487$

Aceste valori pentru  $b^*$  arată că, folosind această euristică,  $A^*$  se va afla în multe situații în care în coada de priorități să fie introduse două sau mai multe stări vecine de la nivelul următor, care vor avea aceeași valoare a euristicii. Mai mult, crește șansa de a găsi noi distanțe scurte de la starea inițială până la anumite stări vizitate care nu mai sunt în coada de priorități (situația în care  $g_{new}(S_j) < g_{old}(S_j)$ ), unde  $S_j$  este o stare vizitată, care nu mai este în coada de priorități).

În acest caz, stările acelea trebuiesc puse din nou în coada de priorități pentru recalcularea noilor valori  $f$ , care estimează costul total trecând prin acele stări, ceea ce aduce un overhead temporal și spațial.

Acest lucru arată un dezavantaj al lui  $A^*$  pentru cazuri mai complexe.

Evoluția exponențială a numărului de stări, duce implicit la evoluția exponențială a timpului de execuție, deoarece complexitatea temporală e determinată de numărul de stări. Acest lucru se poate vedea în graficul de jos.



## Cerința 2

Pentru euristicele h1 și h2 voi prezenta datele sub forma de array-uri numpy. Fiecare array numpy va avea 2 linii și 4 coloane organizate în felul următor.

	Buget=1000	Buget=5000	Buget=10000	Buget=20000
C=0.1				
C=0.5				

Toate valorile au fost obținute prin mediere. Timpul va fi prezentat în milisecunde. Unde există valori nule, înseamnă că rulările folosite (în acest caz, 25 de rulări pentru fiecare combinație) nu au condus la rezultate și nu vor fi luate în considerare.

În `measurements_array_h1` și `measurements_array_h2` stochez măsurătorile obținute în urma MCTS pentru euristica h1, respectiv h2.

Pentru euristica h1

- Pentru case1

Timp

```
In [23]: measurements_array_h1[0][0]*1000
Out[23]: array([[ 341.31479263, 2053.52034776, 2196.59398903, 6648.72006416],
 [ 338.63865852, 281.36119843, 320.47881126, 320.44003487]])
```

lungimea drumului

```
In [24]: measurements_array_h1[0][1]
Out[24]: array([[5., 7.34782609, 6.90909091, 8.76 ],
 [5., 5., 5., 5.]])
```

Numărul de stări

```
In [25]: measurements_array_h1[0][2]
Out[25]: array([[ 3042.5, 15342.30434783, 16842.95454545, 43497.76 ],
 [ 3492.24, 2881.72, 3250.4, 3317.56 ]])
```

- Pentru case2

Timp

```
In [26]: measurements_array_h1[1][0]*1000
Out[26]: array([[ 670.49872875, 5056.60052299, 10969.23367182, 21110.99950473],
 [ 924.99931653, 3477.69379616, 3870.7376252, 3381.60128593]])
```

Lungimea drumului

```
In [27]: measurements_array_h1[1][1]
Out[27]: array([[13.          , 23.          , 24.77777778, 24.06666667],
               [ 8.          , 9.23529412, 8.47826087, 7.88          ]])
```

## Numărul de stări

```
In [28]: measurements_array_h1[1][2]
Out[28]: array([[ 5381.          , 34608.2          , 66704.          ,
                  118521.26666667],
               [ 8888.5          , 30497.17647059, 32661.69565217,
                  28673.04          ]])
```

- Pentru case3

## Time

```
In [29]: measurements_array_h1[2][0]*1000
Out[29]: array([[1217.0009613 , 3208.75027776, 10917.84457366, 22589.06890665],
               [ 0.          , 4001.00549785, 4739.93685842, 13383.51991653]])
```

## Length of road

```
In [30]: measurements_array_h1[2][1]
Out[30]: array([[15.66666667, 13.75          , 23.33333333, 23.42857143],
               [ 0.          , 10.09090909, 10.875          , 11.64          ]])
```

## Numărul de stări

```
In [31]: measurements_array_h1[2][2]
Out[31]: array([[ 9883.66666667, 22179.625          , 64380.16666667,
                  121634.85714286],
               [ 0.          , 36011.          , 42258.1875          ,
                  109931.52          ]])
```

- Pentru case4

## Time

```
In [32]: measurements_array_h1[3][0]*1000
Out[32]: array([[ 467.49961376, 4018.99019877, 18116.66737662, 29397.42616245],
               [ 0.          , 2146.00086212, 0.          , 18394.53577995]])
```

## Length of the road

```
In [33]: measurements_array_h1[3][1]
Out[33]: array([[14.5          , 23.          , 31.22222222, 28.57142857],
               [ 0.          , 13.          , 0.          , 15.          ]])
```

## Numărul de stări

```
In [34]: measurements_array_h1[3][2]
Out[34]: array([[ 3980.75          , 25452.          , 100726.11111111,
                  151946.          ],
               [ 0.          , 19397.          , 0.          ,
                  128147.          ]])
```

Exceptând situațiile unde am avut valori nule, putem constata că, pentru valoarea constantei  $C=0,5$  obținem performanțe în general mai bune, deoarece vor fi explorate mai multe stări din care să putem ajunge la starea finală

Pentru euristica h2, am ales una care numără câte stick-ere sunt incorecte. Un sticker este incorect dacă culorile corespunzătoare fețelor acestuia sunt incorecte, sau dacă sunt corecte sunt orientate greșit.

- Pentru case1

Timp

```
In [35]: measurements_array_h2[0][0]*1000
Out[35]: array([[ 379.0937337 , 2158.06903158, 4968.61105495, 8406.04752586],
 [ 699.42053996, 1113.82616085, 2292.16013908, 1347.03988075]])
```

Lungime drum

```
In [36]: measurements_array_h2[0][1]
Out[36]: array([[ 6.45454545,  9.42857143, 13.55555556, 12.23809524],
 [ 5.          ,  5.26086957,  5.4          ,  5.          ]])
```

Numărul de stări

```
In [37]: measurements_array_h2[0][2]
Out[37]: array([[ 3388.72727273, 16172.          , 33145.11111111, 51683.9047619 ],
 [ 6497.10526316, 10169.          , 19879.64          , 11950.96          ]])
```

- Pentru case2

Timp

```
In [38]: measurements_array_h2[1][0]*1000
Out[38]: array([[ 635.00084877, 4518.88728142, 9228.63049805, 19911.78184748],
 [ 834.00111198, 4049.40162897, 3760.80936477, 5401.56002045]])
```

Lungime drum

```
In [39]: measurements_array_h2[1][1]
Out[39]: array([[ 9.          , 21.5          , 22.375          , 24.375          ],
 [ 7.          ,  8.1          ,  8.33333333,  7.64          ]])
```

Numărul de stări

```
In [40]: measurements_array_h2[1][2]
Out[40]: array([[ 5051.6          , 28864.          , 55084.125          ,
 106743.3125          ],
 [ 7612.4          , 33993.65          , 31756.33333333,
 44424.52          ]])
```

- Pentru case3

Timp

```
In [41]: measurements_array_h2[2][0]*1000
Out[41]: array([[ 0.          , 4601.51425004, 15739.18520321, 26088.2647414 ],
 [ 0.          , 4887.2897625 , 11202.46208631, 15144.2911307 ]])
```

Lungime drum

```
In [42]: measurements_array_h2[2][1]
Out[42]: array([[ 0.          , 18.5          , 27.18181818, 31.94736842],
 [ 0.          ,  9.85714286, 10.69230769, 11.4          ]])
```

## Numărul de stări

```
In [43]: measurements_array_h2[2][2]
Out[43]: array([[ 0.          , 31652.75       , 87532.72727273,
                  134133.05263158],
                [ 0.          , 41379.85714286, 84834.         ,
                  113679.4        ]])
```

- Pentru case4

## Timp

```
In [44]: measurements_array_h2[3][0]*1000
Out[44]: array([[ 0.          , 7056.66637421, 13538.56015205, 22300.69285173],
                [ 0.          , 2691.03264809, 1721.00257874, 25532.31186867]])
```

## Lungime drum

```
In [45]: measurements_array_h2[3][1]
Out[45]: array([[ 0.          , 30.33333333, 29.90909091, 37.         ],
                [ 0.          , 11.         , 13.         , 14.6        ]])
```

## Număr de stări

```
In [46]: measurements_array_h2[3][2]
Out[46]: array([[ 0.          , 39847.         , 70301.18181818,
                  105263.61538462],
                [ 0.          , 22674.         , 14704.         ,
                  166703.8        ]])
```

Și în acest caz, observăm că pentru  $C=0.5$  performanțele sunt, în general mai bune decât pentru cealaltă constantă.

Vom compara euristicele luând, pe rând fiecare metrică:

1. Timpul: cu euristica h1, MCTS tinde să se termine mai repede decât cu euristica h2
2. Lungimea drumului:

Pentru case1: la h1 e prezis corect drumul minim mai des în majoritatea situațiilor

Pentru case2: rezultatele par să urmeze o tendință similară. La h2 s-a înregistrat drumul minim de lungime 7 pentru bugetul=1000 și  $C=0,5$

Pentru case3: pentru  $C=0,1$  lungimile drumurilor obținute sunt mai mici la h1 decât la h2.

Pentru  $C=0,5$ , cu excepția valorii pentru buget 1000, tendințele sunt similare.

3. Numărul de stări: În general, cu euristica h1 se generează mai puține stări decât cu euristica h2. La case2, observăm totuși că pentru  $C=0,1$  se generează mai multe stări pentru euristica h1 decât la h2.

Pentru că euristica  $h_1$  are performanțe în general mai bune (mai ales cu timpul), voi compara rezultatele acesteia cu  $A^*$ . Vom folosi pentru  $A^*$  următoarele rezultate obținute. Indicele 0 e pentru case1, 1 pentru case2, 2 pentru case3 și 3 pentru case4.

```
In [90]: states_a_star[0],solution_length_a_star[0],times_a_star[0]
```

```
Out[90]: (30, 5.0, 11.989831924438477)
```

```
In [91]: states_a_star[1],solution_length_a_star[1],times_a_star[1]
```

```
Out[91]: (109, 7.0, 26.035547256469727)
```

```
In [92]: states_a_star[2],solution_length_a_star[2],times_a_star[2]
```

```
Out[92]: (3145, 9.0, 768.9988613128662)
```

```
In [93]: states_a_star[3],solution_length_a_star[3],times_a_star[3]
```

```
Out[93]: (20900, 11.0, 5645.99871635437)
```

Din perspectiva timpului pentru a ajunge la soluție, acesta este mai mare la MCTS decât la  $A^*$ . Pentru case1, case2, case3 avem timpi mai buni la  $A^*$  decât la MCTS. Pentru case4, avem ca și timp aproximativ 5646 milisecunde, mai mare decât timpii obținuți la MCTS pentru bugetele de 1000 și 5000.

Din perspectiva stărilor, MCTS obține un număr de stări mai mare decât la  $A^*$  pentru toate combinațiile de parametrii MCTS la case1, case2 și case3. La case4, pentru cazul buget=1000 și  $C=0,1$ , respectiv buget=5000 și  $C=0,5$  am obținut un număr de stări mai mic decât la  $A^*$ .

Din perspectiva lungimii drumului, numai la case1 am văzut valori identice sau apropiate de ce am obținut la MCTS.



Referințe:

[1] <https://www.cs.princeton.edu/courses/archive/fall06/cos402/papers/korfrubik.pdf>

[2] Peter Norvig și Stuart J. Russel "Artificial Intelligence: A Modern Approach",  
[https://people.engr.tamu.edu/guni/csce421/files/AI\\_Russell\\_Norvig.pdf](https://people.engr.tamu.edu/guni/csce421/files/AI_Russell_Norvig.pdf), pagina 103