Tema 1 – IA

Student: Prioteasa Cristi Andrei - Grupa: 341 C4

Nota: *In toate datele generate am folosit termenul de conflicte liniare incorect, referindu-ma, de fapt la numarul de inversiuni (pozitiile i,j din Npuzzle pt care i < j => tile[i] > tile[j]).*

Beam Search pentru jocul N-Puzzle

Algoritmul A star

Algorithm configuration	Time mean	Steps mean	States mean	Time variance	Steps variance	States variance
Algorithm: A_STAR Heuristic: Manhattan distance + linear conflicts Problem size:4	2.9913	66.0000	29861.6667	4.3511	168.0000	426737034.8889
Algorithm: A_STAR Heuristic: Manhattan distance + linear conflicts Problem size:5	21.0796	44.6667	83630.6667	353.8471	46.2222	6299262790.8889
Algorithm: A_STAR Heuristic: Manhattan distance + linear conflicts Problem size:6	0.0153	13.8000	118.8000	0.0004	6.9600	24843.7600
Algorithm: A_STAR Heuristic: Manhattan distance Problem size:4	1.5001	31.5000	22565.0000	0.8054	2.2500	298771225.0000
Algorithm: A_STAR Heuristic: Manhattan distance Problem size:5	1.7337	33.6000	72856.0000	4.0113	16.2400	6898198271.6000
Algorithm: A_STAR Heuristic: Manhattan distance Problem size:6	0.0080	13.8000	99.0000	0.0000	6.9600	14372.0000

Algorithm configuration	Solution found
4-easy-Manhattan distance + linear conflicts	60.0%
5-easy-Manhattan distance + linear conflicts	60.0%
6-easy-Manhattan distance + linear conflicts	100.0%
4-easy-Manhattan distance	40.0%
5-easy-Manhattan distance	100.0%
6-easy-Manhattan distance	100.0%

Fiind o cautare exhaustiva in spatiul starilor, ghidata dupa o euristica, limita de memorie impusa per problem size este depasita pentru o parte din stele 4-easy pentru euristica Manhattan distance + 2 * no. inversiuni si Manhattan distance simplu, analog pentru problem size 5.

Pentru a pastra consistenta rezultatelor, in cazul in care A* depaseste memoria alocata per problem size, acesta intoarce esec si nu o solutie partiala.

Numarul de stari pastrate in memorie este mare chiar si pentru problem size 4 (aprox 20 000 – 30 000).

In cazul testului pe problem size 5 cu suma de euristici, varianta timpului este foarte mare deoarece algoritmul gaseste o solutie doar pe 3 din cele 5 teste.

Varianta starilor tinute in memorie este foarte mare pentru fiecare tip de problema deoarece arborele creste exponential (cu factorul de ramificare B), deci si starile tinute in memorie vor creste exponential.

Manhattan distance + numarul de inversiuni nu reprezinta o euristica admisibila, dar obtine rezultate bune pe unele teste (pt 4-easy reuseste sa gaseasca o solutie optima pentru 60% din teste vs 40% pt Manhattan distance)

Algoritmul Beam Search

Pentru algoritmul Beam search rezultatele pentru diferitele configuratii ale parametrilor (beam, euristica, problem-size) pot fi observate in tabelul urmator. Liniile marcate cu rosu sunt instante ale algoritmului in care s-a atins limita de memorie impusa.

Se observa ca euristica Manhattan Distance obtine in medie timpi de rulare mai mici decat Manhattan + no. inversiuni pentru oricare valori ale beam-ului.

Nota: *Datele pot fi consultate mai usor in datasheetul atasat. Celulele in care se afla 0, reprezinta de fapt un numar foarte mic de forma 10^-6 pe care excel nu il reprezinta corect.*

Algorithm configuration	Time mea	Steps mea	States mean	Time varian	Steps variance	States varian
Beam search Heuristic: Manhattan distance + linear conflictsBeam size: 1 Problem si	0.0481			0.0006		
Beam search Heuristic: Manhattan distance + linear conflictsBeam size: 1 Problem si	0.1503			0.0165		
Beam search Heuristic: Manhattan distance + linear conflictsBeam size: 1 Problem si	0.2636	13.7500	14.7500	0.2675	8.6875	8.6875
Beam search Heuristic: Manhattan distance + linear conflictsBeam size: 1 Problem siz	0.0456			0.0007		
lgorithm: Beam search Heuristic: Manhattan distanceBeam size: 1 Problem size:4-eas	0.0304			0.0005		
lgorithm: Beam search Heuristic: Manhattan distanceBeam size: 1 Problem size:5-eas	0.5646			1.0331		
lgorithm: Beam search Heuristic: Manhattan distanceBeam size: 1 Problem size:6-eas	24.2260	55.5000	56.5000	195.5697	2162.2500	2162.2500
gorithm: Beam search Heuristic: Manhattan distanceBeam size: 1 Problem size:4-norm	0.1515			0.0569		
Beam search Heuristic: Manhattan distance + linear conflictsBeam size: 10 Problem s	2.245516491	1384.6	13123.8	8.367515655	1172746.24	104258060.6
Beam search Heuristic: Manhattan distance + linear conflictsBeam size: 10 Problem s	19.38098822	449.5	4390.5	233.407907	111890.25	10939556.25
Beam search Heuristic: Manhattan distance + linear conflictsBeam size: 10 Problem s	0.640747738	240.2	2343.4	1.216931584	158970.16	15745181.84
Beam search Heuristic: Manhattan distance + linear conflictsBeam size: 10 Problem si	0.582792302	743	6948.6	0.306637516	453815.6	40106701.44
gorithm: Beam search Heuristic: Manhattan distanceBeam size: 10 Problem size:4-ea	0.034507656	113	1019	0.000537871	4918.4	446164
gorithm: Beam search Heuristic: Manhattan distanceBeam size: 10 Problem size:5-ea	0.148999214	321.6	3150.4	0.011361676	46738.64	4646828.24
gorithm: Beam search Heuristic: Manhattan distanceBeam size: 10 Problem size:6-ea	0.003989649	13.8	90	2.00663E-06	6.96	799.2
orithm: Beam search Heuristic: Manhattan distanceBeam size: 10 Problem size:4-nor	0.052457523	167.8	1461.4	0.000279493	1820.56	139720.24
orithm: Beam search Heuristic: Manhattan distanceBeam size: 10 Problem size:5-nor	0.459307671	835	8211.2	0.080570475	182501.6	18246311.76
orithm: Beam search Heuristic: Manhattan distanceBeam size: 10 Problem size:6-nor	117.1997808	7467.8	74533.6	19162.45929	14665183.76	1464153069
Beam search Heuristic: Manhattan distance + linear conflictsBeam size: 50 Problem s	1.028837919	332.6	13265	0.542846528	45301.44	82422246.8
Beam search Heuristic: Manhattan distance + linear conflictsBeam size: 50 Problem s	25.51319466	35	1124	160.7546147	0	0
Beam search Heuristic: Manhattan distance + linear conflictsBeam size: 50 Problem s	6.158817911	17.5	269.75	148.4547217	30.25	46621.1875
Beam search Heuristic: Manhattan distance + linear conflictsBeam size: 50 Problem si	0.780188465	240.2	10146.8	0.5383512	38171.76	86276963.76
gorithm: Beam search Heuristic: Manhattan distanceBeam size: 50 Problem size:4-ea	0.088961077	84.6	2716.4	0.00324751	1137.44	2593587.84
gorithm: Beam search Heuristic: Manhattan distanceBeam size: 50 Problem size:5-ea	0.2746768	134.4	6043	0.081873774	14494.64	35737534
gorithm: Beam search Heuristic: Manhattan distanceBeam size: 50 Problem size:6-ea	0.007016242	13.8	157.4	9.82878E-06	6.96	5481.04
orithm: Beam search Heuristic: Manhattan distanceBeam size: 50 Problem size:4-nor	0.100207233	115.4	3098.8	0.000479836	429.04	409756.56
orithm: Beam search Heuristic: Manhattan distanceBeam size: 50 Problem size:5-nor	0.465411329	231.4	10283.2	0.022836173	4027.04	10184489.76
orithm: Beam search Heuristic: Manhattan distanceBeam size: 50 Problem size:6-nor	89.0818346	2749	136688.2	15451.6304	1418534.8	3557044501
Beam search Heuristic: Manhattan distance + linear conflictsBeam size: 100 Problem :	1.094445038	193	14310.2	0.581754554	9123.2	95601175.76
Beam search Heuristic: Manhattan distance + linear conflictsBeam size: 100 Problem :	24.33443283	35	1901	144.7902396	0	0
Beam search Heuristic: Manhattan distance + linear conflictsBeam size: 100 Problem :	0.123206043	19	508.4	0.013870727	33.2	219517.04
eam search Heuristic: Manhattan distance + linear conflictsBeam size: 100 Problem si	0.621405649	127.8	8407.4	0.128233107	2232.56	23170162.64
gorithm: Beam search Heuristic: Manhattan distanceBeam size: 100 Problem size:4-ea	0.085171938	67.8	2591.4	0.002171666	347.36	1792378.24
gorithm: Beam search Heuristic: Manhattan distanceBeam size: 100 Problem size:5-ea	0.287613305	82.4	6514.2	0.113119852	5385.04	52194341.76
gorithm: Beam search Heuristic: Manhattan distanceBeam size: 100 Problem size:6-ea	0.006583548	13.8	158.6	1.25713E-05	6.96	5740.24
orithm: Beam search Heuristic: Manhattan distanceBeam size: 100 Problem size:4-nor	0.149399396	110.6	4142.2	0.001606233	327.44	1119498.16
orithm: Beam search Heuristic: Manhattan distanceBeam size: 100 Problem size:5-nor	0.913691139	216.2	18239.6	0.056041538	1832.16	13702181.84
orithm: Beam search Heuristic: Manhattan distanceBeam size: 100 Problem size:6-nor	58.05446849	1534.2	151449	4430.721585	1152246.96	11548962143
gorithm: Beam search Heuristic: Manhattan distanceBeam size: 500 Problem size:4-ea	0.118882132	70.6	3466.4	0.007055648	458.24	5422183.84
gorithm: Beam search Heuristic: Manhattan distanceBeam size: 500 Problem size:5-ea	0.357993277	43.6	8313	0.024036035	54.64	11891482
gorithm: Beam search Heuristic: Manhattan distanceBeam size: 500 Problem size:6-ea	0.007728934	13.8	158.6	1.05814E-05	6.96	5740.24
orithm: Beam search Heuristic: Manhattan distanceBeam size: 500 Problem size:4-nor	0.184104061	109.8	5318	0.005209894	284.56	3822713.2
orithm: Beam search Heuristic: Manhattan distanceBeam size: 500 Problem size:5-nor	2.903456211	157.8	53261.4	0.0353309	107.36	28168720.24
orithm: Beam search Heuristic: Manhattan distanceBeam size: 500 Problem size:6-nor	16.59172416	466.2	217375.6	119.9339618	56122.96	13736466971
orithm: Beam search Heuristic: Manhattan distance + linear conflictsBeam size: 500 Problem size:5-normal						
orithm: Beam search Heuristic: Manhattan distance + linear conflictsBeam size: 500 Problem size:6-normal						

Algorithm configuration	Solution found
4-easyManhattan distance + linear conflicts 1	0.0%
5-easyManhattan distance + linear conflicts 1	0.0%
6-easyManhattan distance + linear conflicts 1	80.0%
4-normalManhattan distance + linear conflicts 1	0.0%
4-easyManhattan distance 1	0.0%
5-easyManhattan distance 1	0.0%
6-easyManhattan distance 1	40.0%
4-normalManhattan distance 1	0.0%
4-easyManhattan distance + linear conflicts 10	100.0%
5-easyManhattan distance + linear conflicts 10	40.0%
6-easyManhattan distance + linear conflicts 10	100.0%
4-normalManhattan distance + linear conflicts 10	100.0%
4-easyManhattan distance 10	100.0%
5-easyManhattan distance 10	100.0%
6-easyManhattan distance 10	100.0%
4-normalManhattan distance 10	100.0%
5-normalManhattan distance 10	100.0%
6-normalManhattan distance 10	100.0%
4-easyManhattan distance + linear conflicts 50	100.0%
5-easyManhattan distance + linear conflicts 50	20.0%
6-easyManhattan distance + linear conflicts 50	80.0%
4-normalManhattan distance + linear conflicts 50	100.0%
4-easyManhattan distance 50	100.0%
5-easyManhattan distance 50	100.0%
6-easyManhattan distance 50	100.0%
4-normalManhattan distance 50	100.0%
5-normalManhattan distance 50	100.0%
6-normalManhattan distance 50	100.0%
4-easyManhattan distance + linear conflicts 100	100.0%
5-easyManhattan distance + linear conflicts 100	20.0%
6-easyManhattan distance + linear conflicts 100	100.0%
4-normalManhattan distance + linear conflicts 100	100.0%
4-easyManhattan distance 100	100.0%
5-easyManhattan distance 100	100.0%
6-easyManhattan distance 100	100.0%
4-normalManhattan distance 100	100.0%
5-normalManhattan distance 100	100.0%
6-normalManhattan distance 100	100.0%
4-easyManhattan distance + linear conflicts 500	100.0%
5-easyManhattan distance + linear conflicts 500	100.0%
6-easyManhattan distance + linear conflicts 500	100.0%
4-normalManhattan distance + linear conflicts 500	100.0%
4-easyManhattan distance 500	100.0%
5-easyManhattan distance 500	100.0%
6-easyManhattan distance 500	100.0%
4-normalManhattan distance 500	100.0%
5-normalManhattan distance 500	100.0%
6-normalManhattan distance 500	100.0%

Deoarece Beam Search incearca diferite cai, media timpului de rulare pana la gasirea unei solutii este mult mai mic decat la A* (e.g: 2.99 s pe 4-easy versus 0.08 s). Si media numarului de stari tinute in memorie este mai mic, deorece in dictionarul de vizitat se adauga la un moment dat maxim B * 4 vecini noi.

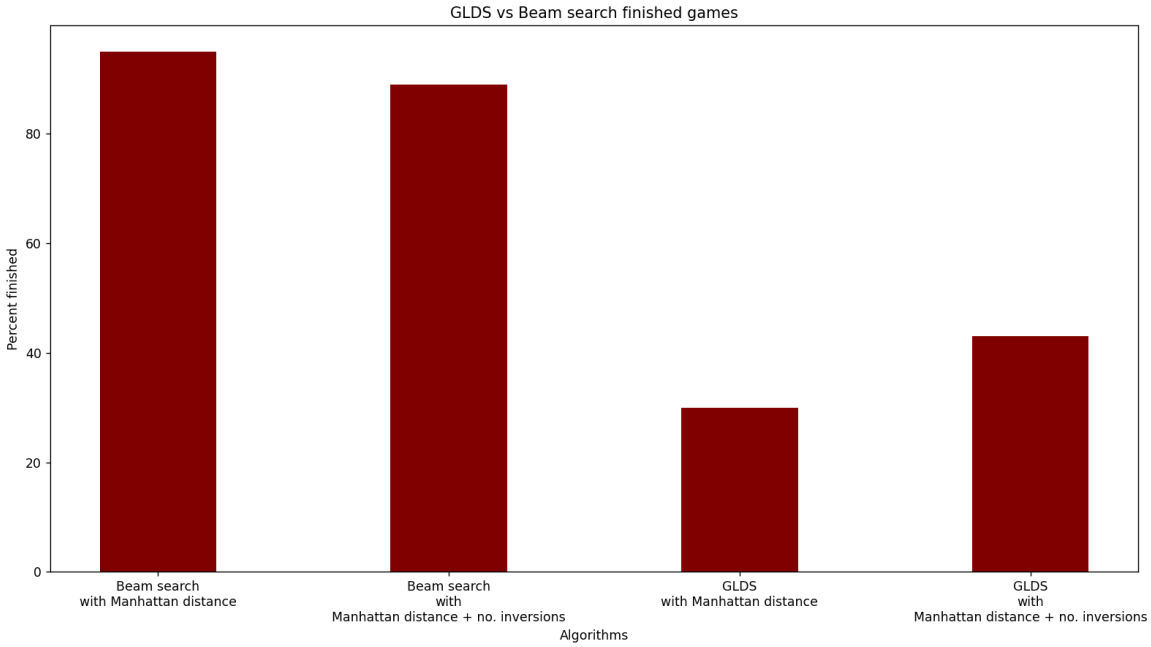
Timpul de rulare creste o data cu dimensiunea beam-ului si dimensiunea problemei, asemenea numarul de stari tinute in memorie. Varianta timpulu si varianta lungimii caii gasite la beam search tinde catre varianta lungimii caii gasite in cazul algoritmului A* (respectiv varianta timpului) cu cat beam-ul este mai mare. Acest lucru are sens pentru ca daca Beam -> infinit, atunci Beam search devine un A*.

Algoritmul A* gaseste solutia optima, dar ocupa multe stari in memorie si ia un timp mai indelungat. Algoritmul Beam Search cu un Beam mic (100 – 500) face un compromis intre optimalitatea solutiei si viteza de cautare / numarul de stari tinute in memorie.

De exemplu pentru problema 4 easy folosind euristica Manhattan distance + no inversiuni, A* obtine o cale optime de 66 pasi, pe cand Beam search gaseste solutii de lungime 1384 cu beam 10, 332 cu Beam 50 si 193 cu Beam 100. Estimez ca pentru un Beam 1000 se va gasi o solutie cu un numar de pasi comparabil cu cel al A*, dar folosind mai putine resurse si un timp mai scurt de rulare. In cele mai multe cazuri o solutie suboptima, dar apropiata de cea reala cu o marja de eroare epsilon este suficienta pentru majoritatea aplicatiilor practice, deci un Beam Search ar fi de preferat.

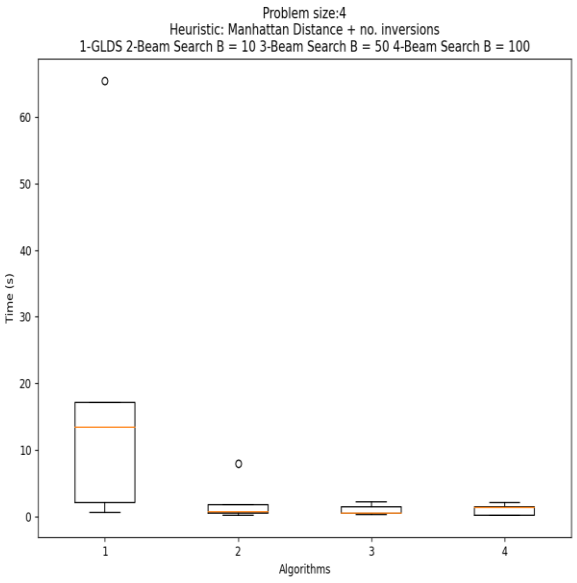
Generalized Limited Discrepancy Search pentru jocul N-puzzle

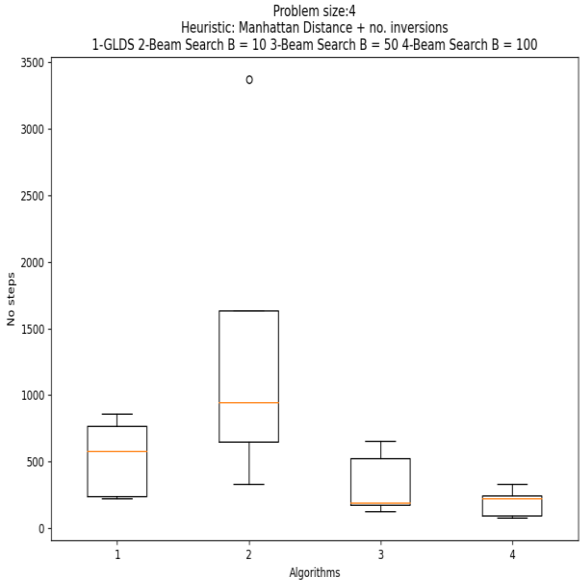
Am decis sa implementez cazul de baza, cand discrepancy = 0 in mod iterativ. Chiar si in acest mod, GLDS nu gaseste solutii in cel putin 50% din testele propuse cu oricare din cele doua euristici. Se observa totusi ca euristica Manhattan Distance + no. Inversions gaseste cu 13% mai multe solutii decat Manhattan distance si intr-un timp mai bun, desi pe testele problemei 4-normal, Manhattan Distance gaseste solutii intr-un timp de 100x mai bun ca Manhattan distance + no. Inversions.

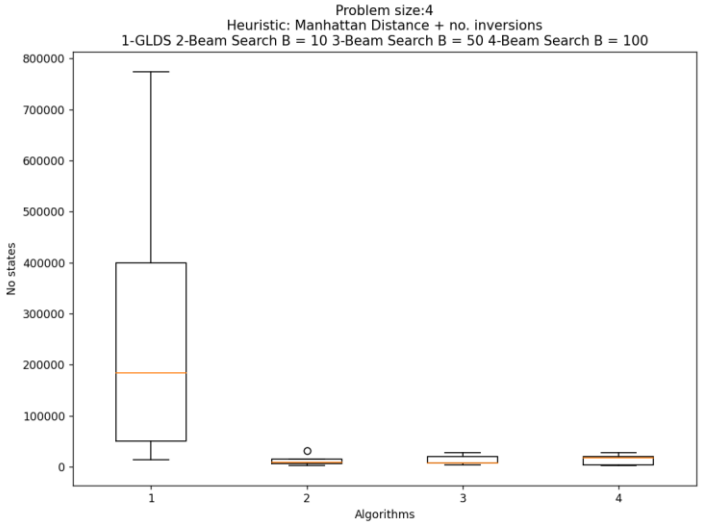


Problem size: 4-easy

Manhattan Distance + no. inversions

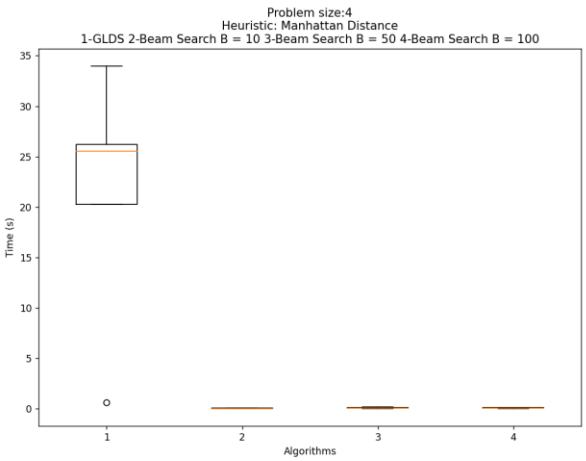


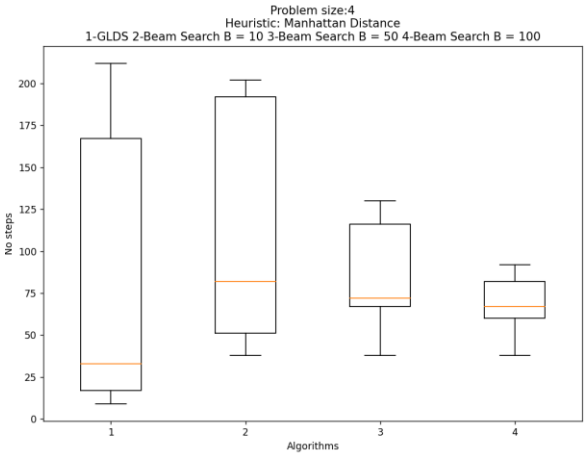


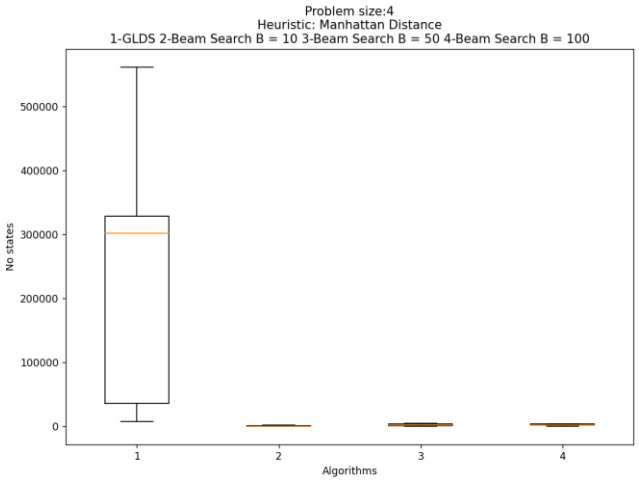


Evident timpul de rulare pentru GLDS este mult mai mare decat variantele cu Beam search, chiar si pt beam mare. Numarul de pasi pana la solutie este mai mare decat la Beam search. (facand algoritmul iterativ, numarul de pasi a fost numarat intre discrepante incercate). Numarul de stari tinute in memorie >> numarul de stari tinute in memorie de catre orice varianta de Beam Search (GLDS iterativ).

Manhattan Distance





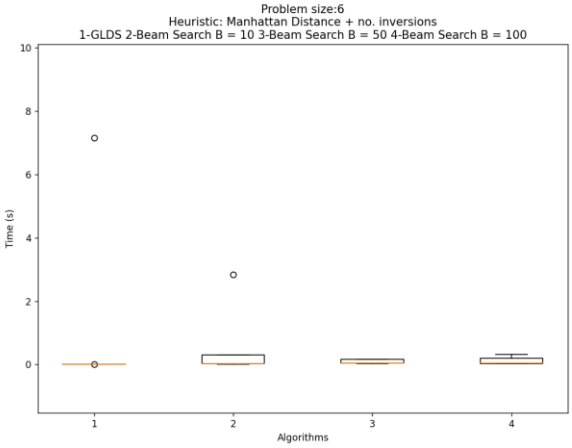


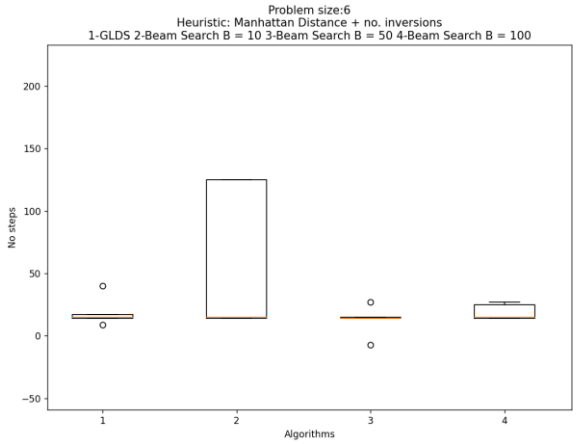
Numarul de pasi pana la solutie, asemanator cu cel de la un Beam search cu b = 10, este totusi mai mare ca cel de la GLDS folosind euristica Manhattan Distance + no.inversiuni. Timpul de rulare este de peste 100x mai mare ca la Beam Search, iar numarul de stari tinute in memorie este extrem de mare.

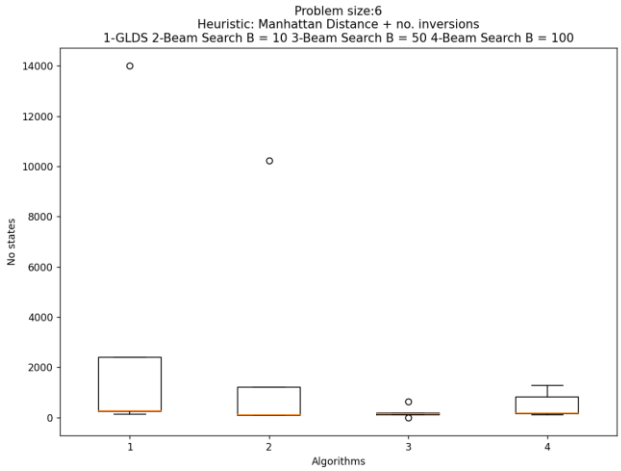
Nota: *am ales sa impun limita 1 000 000 pe stari pentru orice problem size, altfel nu ar fi trecut nici macar un test cu euristica mea*

Problem size: 6-easy

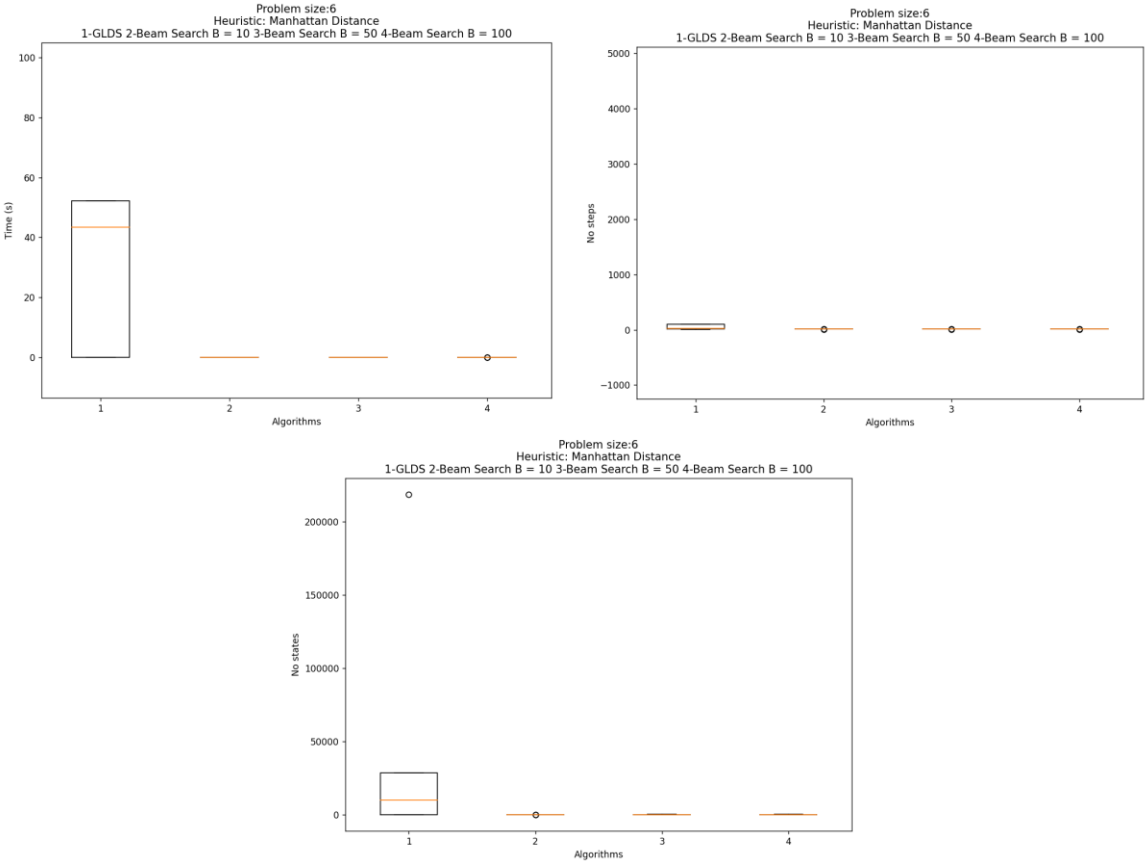
Manhattan Distance + no. inversions







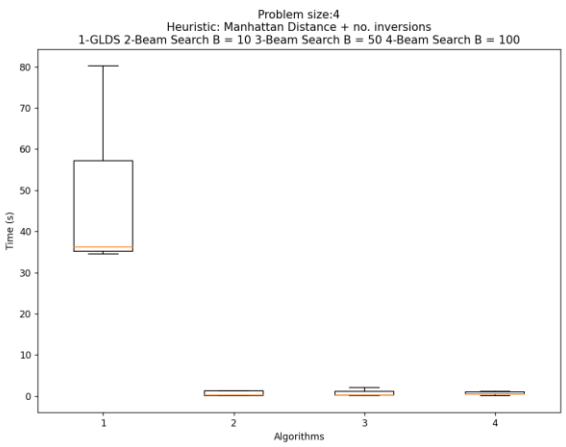
Manhattan Distance

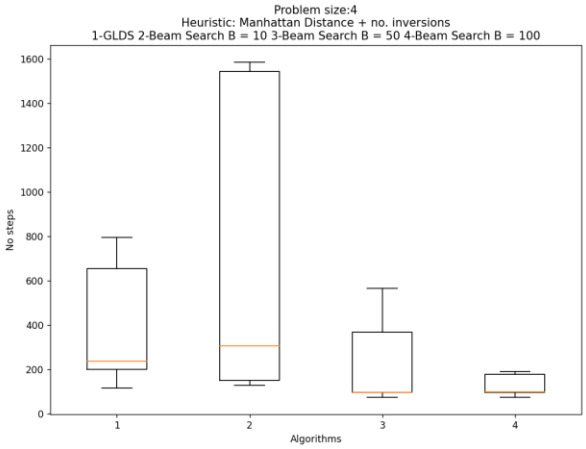


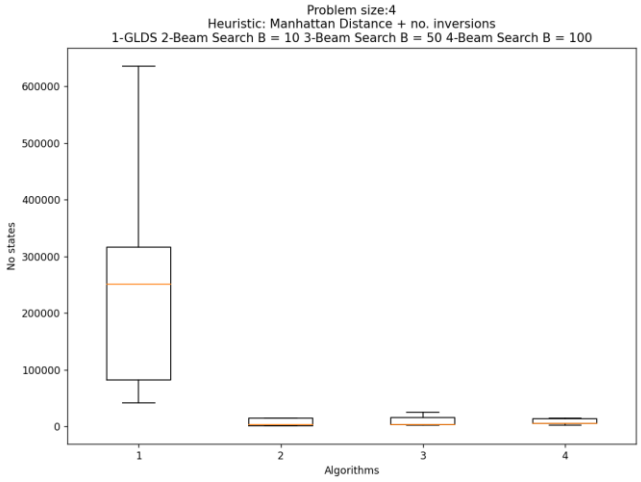
In cazul 6-easy fiind teste usoare, rezultatele sunt comparabile in privinta numarului de pasi pana la solutie, dar timpul de executie si numarul de stari este cu acelasi ordin de marime mai mare ca la testele precedente fata de Beam Search.

Problem size: 4-normal

Manhattan Distance + no.inversions

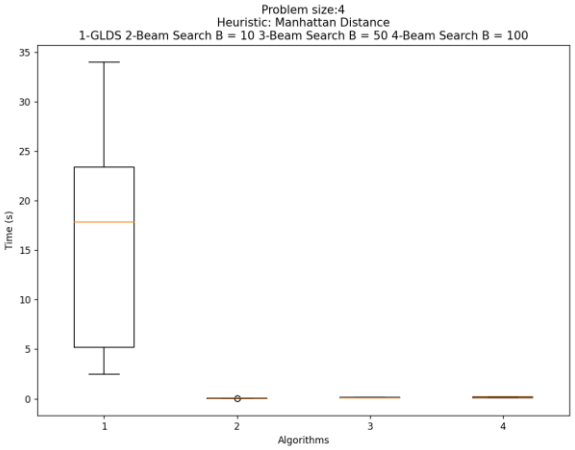


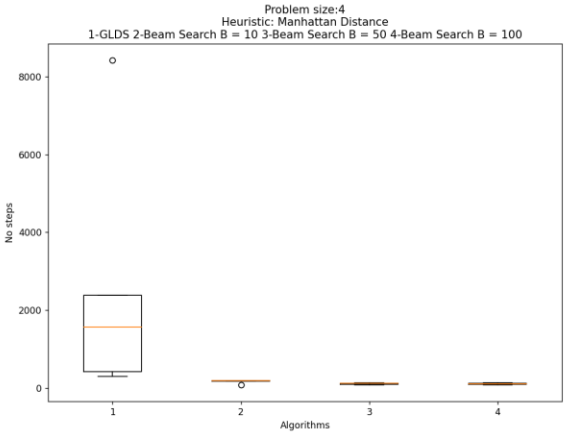


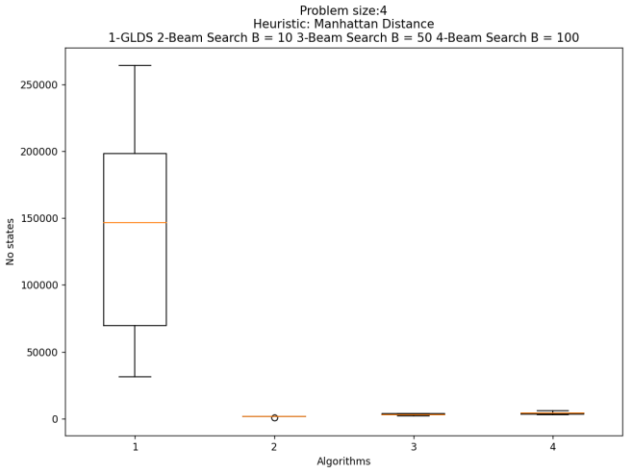


Din forma box-ului se observa ca timpul de rulare pentru majoritatea testelor 4-normal cu GLDS este in jur de 35 secunde, crescand apoi cu dificulteatea testelor. Timpul de rulare pentru toate instantele prezente de Beam Search este sub 5 secunde. Numarul de pasi pana la gasirea solutiei este doar mai mic ca un Beam Search cu b = 10, care probabil ia o cale gresita si o expandeaza pana ajunge la o solutie, fiind foarte lunga. Numarul de stari tinute in memorie (GLDS iterativ) este in medie 250 000, cu multe ordine de marime peste Beam Search.

Manhattan Distance







In afara de timpul de rulare mult mai mic ca GLDS folosind euristica Manhattan Distance + no. Inversiuni, numarul de pasi este mult mai mic iar in medie se tin 150 000 stari in memorie, deci cu 40% mai putin.

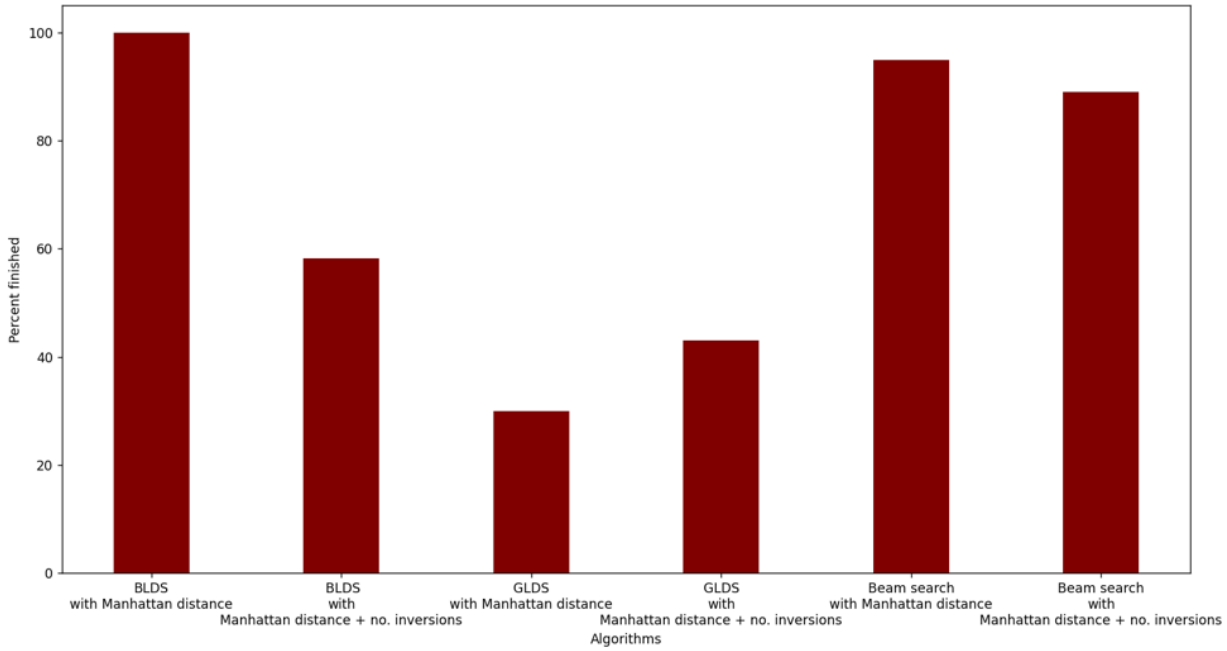
Algorithm configuration	Percent solution found	Percent process crashed unexpectedly	Percent memory exceeded
GLDS with manhattan + num inversions problem size: 4-easy	100%	0%	
GLDS with manhattan + num inversions problem size: 4-normal	40%	0%	60%
GLDS with manhattan + num inversions problem size: 6-easy	80%	20%	
GLDS with manhattan + num inversions problem size: 6-normal	0%	100%	0%
GLDS with manhattan + num inversions problem size: 5-easy	20%		
GLDS with manhattan + num inversions problem size: 5-normal	0%	100%	0%
GLDS with manhattan problem size: 4-easy	80%		20%
GLDS with manhattan problem size: 4-normal	40%		60%
GLDS with manhattan problem size: 6-easy	80%		20%
GLDS with manhattan problem size: 6-normal	0		
GLDS with manhattan problem size: 5-easy	0		
GLDS with manhattan problem size: 5-normal	0		

Beam Search cu Limited Discrepancy Backtracking (BLDS) pentru jocurile N-puzzle si Turnurile din Hanoi

Rezultate NPuzzle

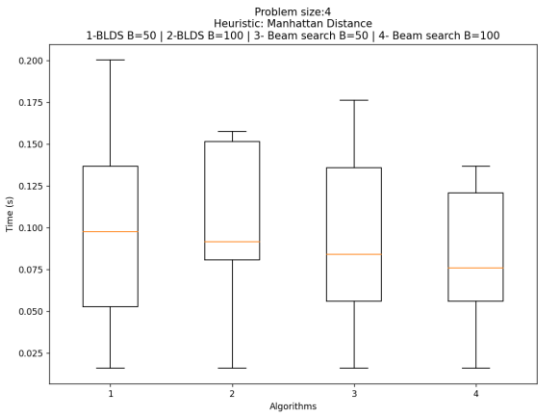
Heuristic	Time mean	Steps mean	States mean	Time variance	Steps variance	States variance	Problem size	Beam
Manhattan	0.119170046	70.6	3500.4	0.006944939	458.24	5453771.44	4-easy	500
Manhattan	0.370140791	43.6	9837	0.026764548	54.64	19488247.2	5-easy	500
Manhattan	0.00638299	13.8	164.6	1.30E-05	6.96	5932.24	6-easy	500
Manhattan	0.208652687	109.8	5371.8	0.006387057	284.56	3852974.16	4-normal	500
Manhattan	2.927232218	157.8	66354	0.096412589	107.36	45881010.8	5-normal	500
Manhattan	20.78635359	513.5	335833	173.5704888	58967.25	28529645256	6-normal	500
Manhattan	0.099529028	67.8	2700.8	0.002695336	347.36	2002698.16	4-easy	100
Manhattan	0.32890172	82.4	8569.8	0.159525109	5385.04	94123750.96	5-easy	100
Manhattan	0.006585026	13.8	164.6	1.06E-05	6.96	5932.24	6-easy	100
Manhattan	0.161445808	110.6	4348	0.002724997	327.44	1350826.4	4-normal	100
Manhattan	0.953983307	216.2	23287	0.053616874	1832.16	25840002.4	5-normal	100
Manhattan	27.63292956	1742	247271	548.5244436	1224404.5	26096522699	6-normal	100
Manhattan	0.100730419	84.6	2962.6	0.004144628	1137.44	3160335.44	4-easy	50
Manhattan	0.312066746	134.4	8150.6	0.120416271	14494.64	65930903.44	5-easy	50
Manhattan	0.007978773	13.8	165.8	1.54E-05	6.96	6205.359999	6-easy	50
Manhattan	0.123645115	115.4	3339	0.001119365	429.04	515864.8	4-normal	50
Manhattan	0.53759613	231.4	13209.8	0.043330209	4027.04	18916754.56	5-normal	50
Manhattan	10.70920388	1853.333333	133054.6667	2.196546272	22137.55556	95525848.22	6-normal	50
Manhattan	0.040890503	113	1241.2	0.000888022	4918.4	652100.56	4-easy	10
Manhattan	0.169347048	321.6	4571.8	0.016653725	46798.64	9798286.16	5-easy	10
Manhattan	0.00399003	13.8	117.2	2.78E-06	6.96	1760.56	6-easy	10
Manhattan	0.055562925	167.8	1742.2	0.000301785	1820.56	212029.36	4-normal	10
Manhattan	0.548705816	835	11606	0.122506613	182501.6	37539484	5-normal	10
Manhattan							6-normal	10

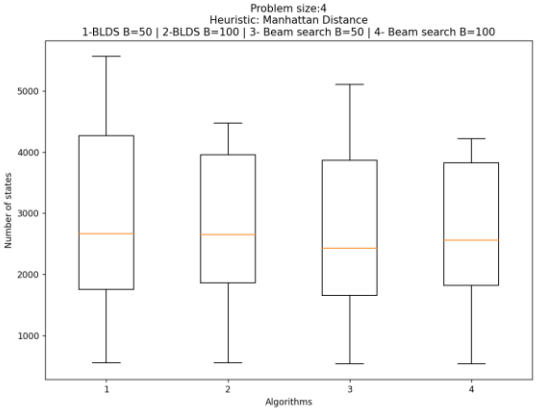
Heuristic	Time mean	Steps mean	States mean	Time variance	Steps variance	States variance	Problem size	Beam
Manhattan + no. inversions	1.141039753	119.8	14858.4	0.713798487	914.56	112779745.8	4-easy	500
Manhattan + no. inversions	1.625398874	46.5	13408.5	1.124543241	132.25	80757182.25	5-easy	500
Manhattan + no. inversions	0.202949095	19	892.2	0.064905543	33.2	1358161.76	6-easy	500
Manhattan + no. inversions	1.846671486	123	22966.8	2.684591598	2176.4	393162282.6	4-normal	500
Manhattan + no. inversions							5-normal	500
Manhattan + no. inversions							6-normal	500
Manhattan + no. inversions	1.187683392	193	15150.8	0.707983777	9123.2	109824329	4-easy	100
Manhattan + no. inversions							5-easy	100
Manhattan + no. inversions	0.118883991	19	580	0.012919322	33.2	328753.6	6-easy	100
Manhattan + no. inversions	0.652939701	127.8	8888.4	0.138274741	2232.56	25308203.84	4-normal	100
Manhattan + no. inversions							5-normal	100
Manhattan + no. inversions							6-normal	100
Manhattan + no. inversions	1.143004131	332.6	14190.8	0.727444749	45301.44	94438897.36	4-easy	50
Manhattan + no. inversions							5-easy	50
Manhattan + no. inversions	0.037233829	14.33333333	152	5.39E-05	0.222222222	658.6666667	6-easy	50
Manhattan + no. inversions	0.887528515	240.2	10946	0.717135545	38171.76	98432727.6	4-normal	50
Manhattan + no. inversions							5-normal	50
Manhattan + no. inversions							6-normal	50
Manhattan + no. inversions	0.561208407	639.6666667	7426	0.064384707	62662.88889	8573150	4-easy	10
Manhattan + no. inversions	0.696298122	449.5	6040	0.285936128	111890.25	21325924	5-easy	10
Manhattan + no. inversions	0.695145178	240.2	3387	1.464562272	158970.16	33472725.2	6-easy	10
Manhattan + no. inversions	0.69925437	743	8303.2	0.4613606	453815.6	56826086.96	4-normal	10
Manhattan + no. inversions							5-normal	10
Manhattan + no. inversions							6-normal	10

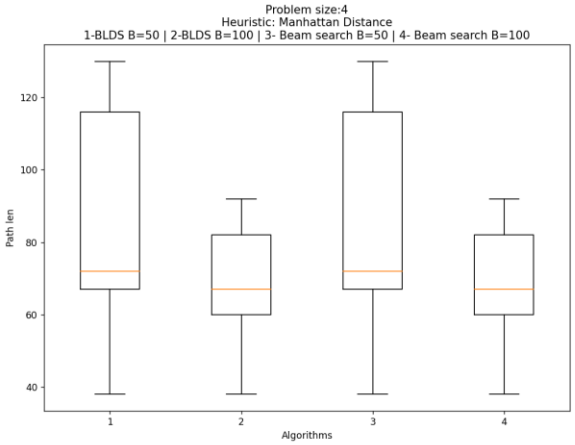


Nota: *Restul graficelor pot fi generate cu ajutorul scheletului de cod pentru euristica si configuratia dorita. Am inclus aici doar graficele pt BLDS vs Beam Search cu Manhattan Distance si B = [50,100]*

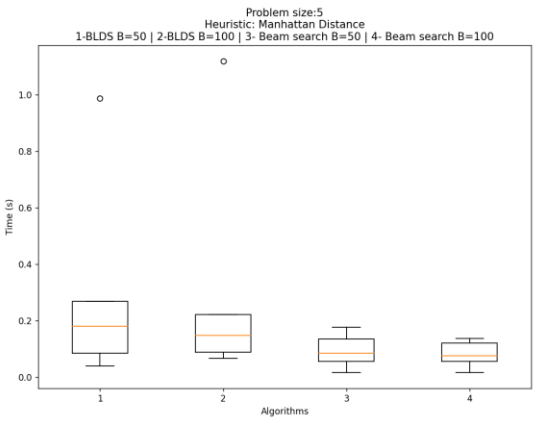
Problem size 4 easy with Manhattan Distance

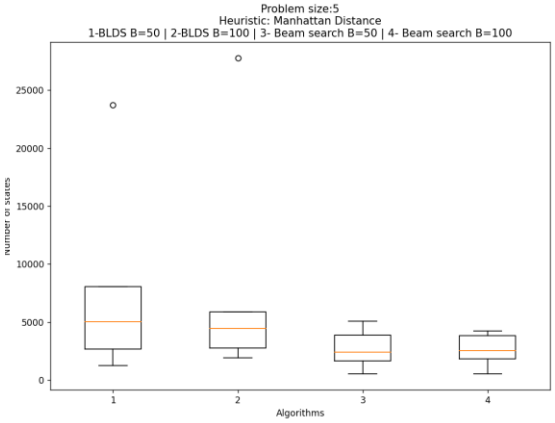


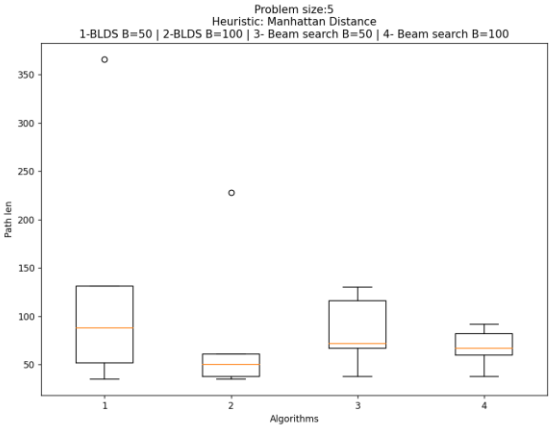




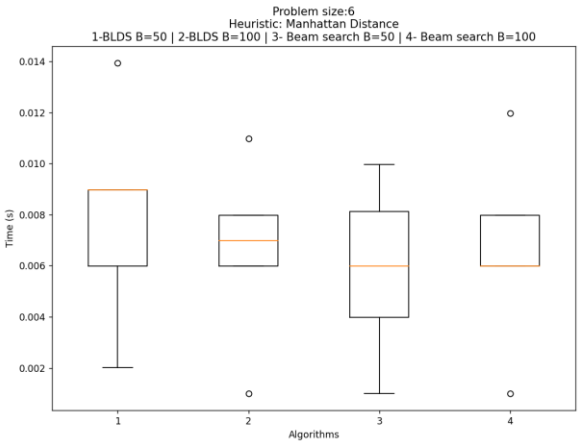
Problem size 5 easy with Manhattan Distance

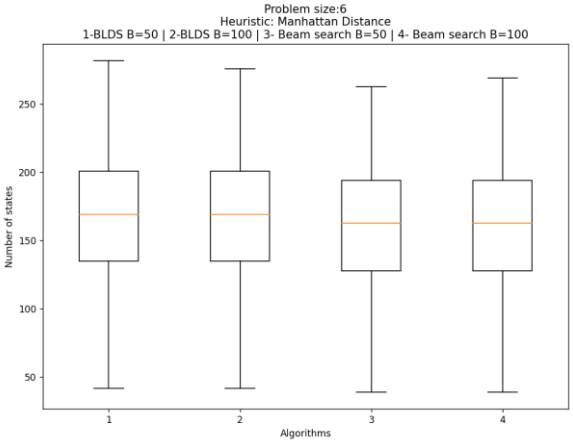


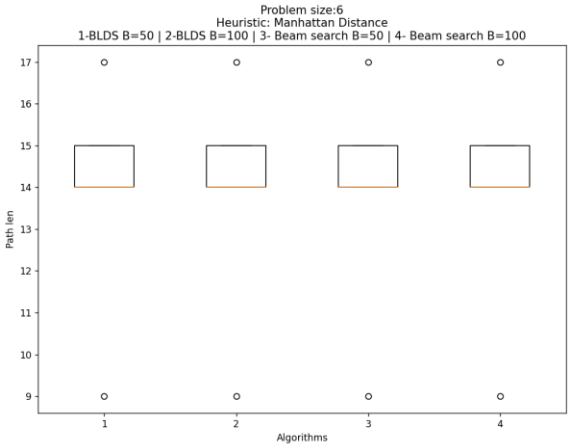




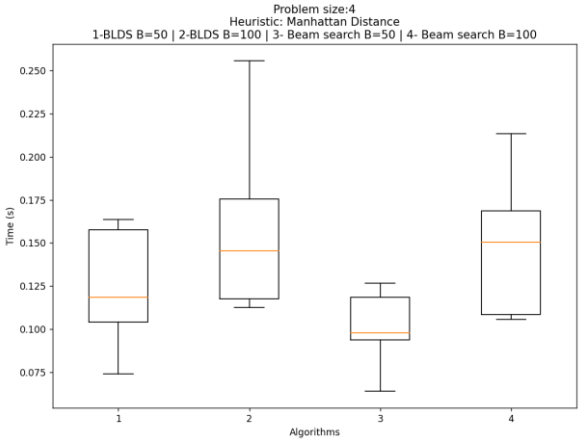
Problem size 6 easy with Manhattan Distance

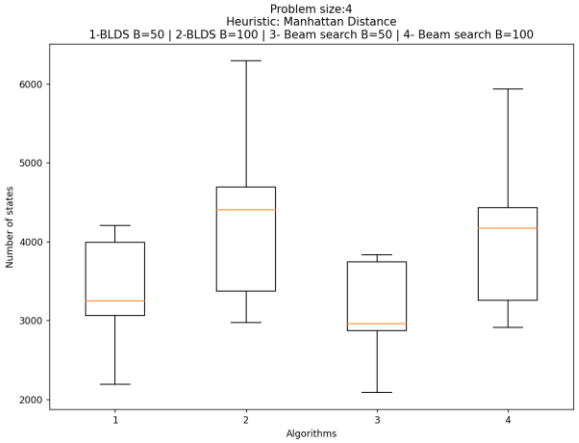


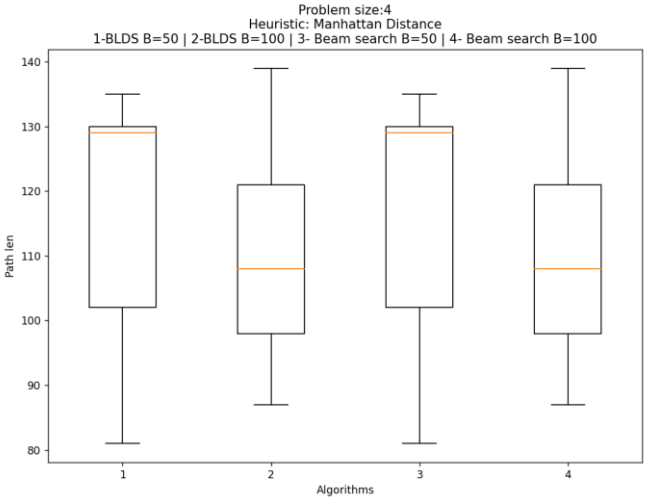




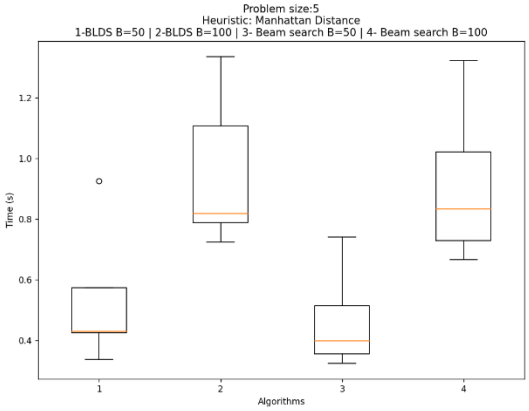
Problem size 4 normal with Manhattan Distance

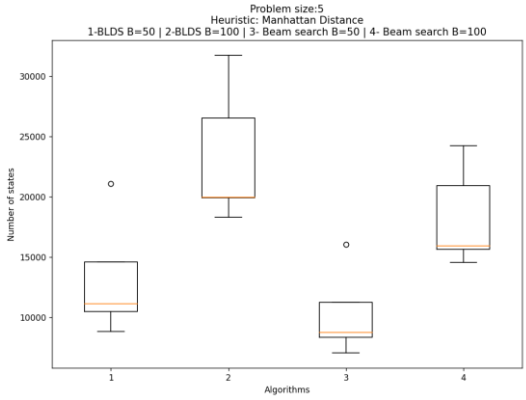


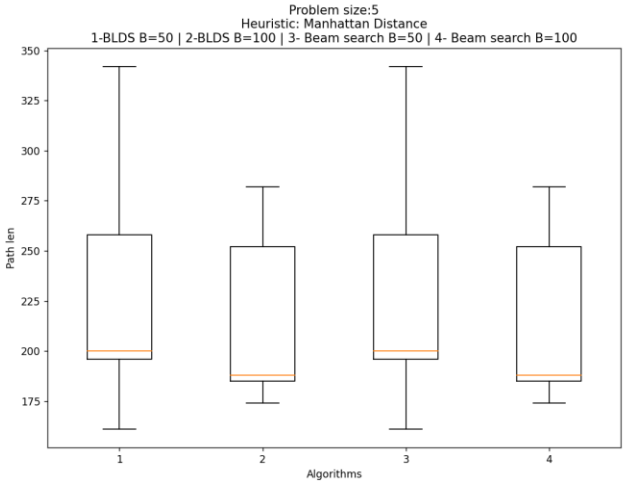




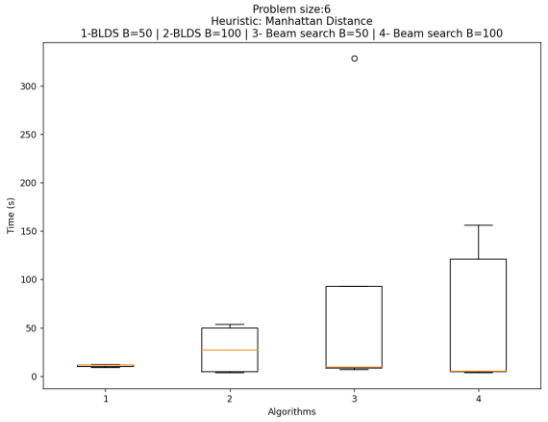
Problem size 5 normal with Manhattan Distance

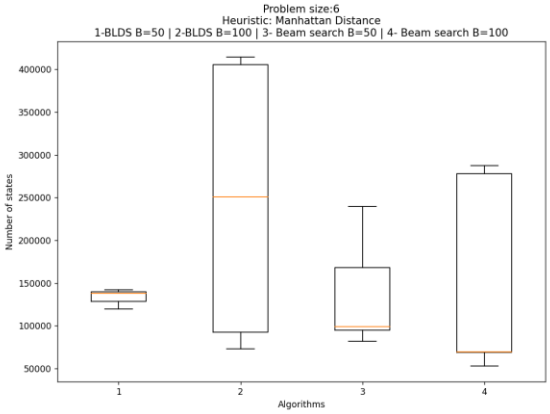


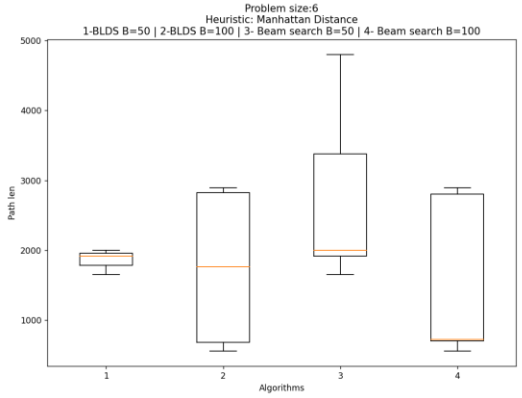




Problem size 6 normal with Manhattan Distance







Concluzie: *Pentru testele 4-easy, 5-easy,6-easy, 4-normal, 5-normal rezultatele obtinute te Beam Search si BLDS sunt comparabile. Pentru problema 6-normal se poate observa ca BLDS performeaza mult mai bine, cu un timp de cautare mult mai bun si mult mai putine stari in memorie decat Beam Search pentru acelasi beam.*

Turnurile din Hanoi

Implementarea jocului Turnurile din Hanoi se bazeaza pe formalismul explicat aici [1]. https://aries.ektf.hu/~gkusper/ArtificialIntelligence_LectureNotes.v.1.0.4.pdf

Euristica folosita reprezinta numarul de misplaced tiles.

Nota: *Liniile marcate cu rosu reprezinta instante ale problemei in care s-a atins limita maxima de memorie impusa*.

Number of dis	Steps to	States used	Time used	Beam
4	11	209	0.007979631	50
5	23	965	0.034914494	50
6	59	3530	0.139626265	50
7	100	7932	0.79488945	50
8	857	59439	2.481371164	50
9	470	43689	5.535364866	50
10	Crash	Crash	Crash	50
11	Crash	Crash	Crash	50
12	Crash	Crash	Crash	50
13	Crash	Crash	Crash	50
14	Crash	Crash	Crash	50
15	Crash	Crash	Crash	50

Number of dis	Steps to	States used	Time used	Beam
4	11	209	0.007977724	100
5	20	809	0.034908533	100
6	38	3918	0.167551517	100
7	77	10397	0.394292116	100
8	208	26984	2.732302904	100
9	635	100517	53.90506721	100
10	Crash	Crash	Crash	100
11	Crash	Crash	Crash	100
12	Crash	Crash	Crash	100
13	Crash	Crash	Crash	100
14	Crash	Crash	Crash	100
15	Crash	Crash	Crash	100

Number of disk	Steps to solution	States used	Time used	Beam
4	11	209	0.008013487	500
5	20	809	0.035890341	500
6	28	2761	0.158594608	500
7	50	13291	0.752188206	500
8	110	51312	2.63278389	500
9	781	238501	53.76115847	500
10	750	362708	227.5937729	500
11	MEM	1000000	450.6400502	500
12	MEM	1000000	95.23199463	500
13	MEM	1000000	97.72797465	500
14	MEM	1000000	105.5012724	500
15	MEM	1000000	113.3905926	500

Number of disk	Steps to solution	States used	Time used	Beam
4	7	209	0.007954597	1000
5	13	809	0.035906792	1000
6	28	2761	0.157577753	1000
7	40	11426	0.643278599	1000
8	72	51675	2.737843275	1000
9	170	115728	7.133862257	1000
10	1006	730928	141.505625	1000
11	MEM	1000000	438.2710462	1000
12	MEM	1000000	93.03755474	1000
13	MEM	1000000	98.58181787	1000
14	MEM	1000000	101.20418	1000
15	MEM	1000000	110.4963353	1000

Se poate observa ca pentru valori mici ale beam-ului se gaseste o solutie suboptima doar pentru instantele problemei cu numarul de discuri <= 9, iar pentru B = 500,1000 se gaseste o solutie in memoria alocata si pentru N = 10.

Nu se gaseste solutia optima, cu numarul de pasi $2^{nr\_discuri}$ -1 (formula care sper ca se pastreaza si daca numarul de stive este egal cu 4) pentru nicio dimensiune a beam-ului pentru probleme cu un numar de discuri mai mare ca 6.