CSCE 625: ARTIFICIAL INTELLIGENCE: PROGRAMMING ASSIGNMENT 1

- ANIKET SANJIV BONDE

Performance Metrics Table

1) These are the averaged results for my best heuristic. (Averaged over 10 random starts)

Stacks	Blocks	Total Iterations	Max Queue Size	Depth	Time Taken (in seconds)	Number of Failures
3	5	9	18	8	0.003	0
	6	16	51	13	0.01	0
	7	30	114	19	0.02	0
	8	40	121	22	0.02	0
	9	41	140	22	0.01	0
	10	98	222	26	0.1	0
	11	200	1030	25	0.02	1
	12	221	1203	38	1.02	3
	13	552	3023	44	0.98	2
	14	568	4023	48	1.76	4
	15	662	5043	45	1.54	4
	16	876	9065	76	3.04	7
4	6	11	65	8	0.02	0
	7	15	80	11	0.02	0
	8	15	90	13	0.04	0
	9	22	154	19	0.3	0
	10	37	233	20	0.09	0
	11	60	321	31	0.2	0
	12	99	662	27	0.534	0
	13	101	701	28	0.34	0
	14	109	977	36	0.9	1
	15	112	1012	39	1.2	3
	16	143	1509	44	1.4	2

Stacks	Blocks	Total Iterations	Max Queue Size	Depth	Time Taken (in seconds)	Number of Failures
5	10	14	165	14	0.03	0
	11	35	1034	22	0.08	0
	12	135	431	26	0.09	0
	13	63	987	25	0.2	0
	14	98	976	29	0.1	0
	15	142	1497	35	0.9	0
	16	168	2054	42	0.2	1
6	11	65	234	31	0.52	0
	12	98	343	34	0.64	0
	13	154	1834	43	0.74	0
	14	198	2531	47	0.9	1
	15	167	3133	54	1.2	0
	16	243	3231	64	1.44	1
7	11	60	232	31	0.2	0
	12	93	234	27	0.53	0
	13	98	701	28	0.4	0
	14	103	1023	36	0.9	1
	15	142	1932	39	1.2	2
	16	196	2283	54	1.54	0

Discussion about results and failures:

- The simple heuristic's results are too far worse than this new modified heuristic.
- Almost the factor of improvement over the base heuristic is 50.
- Even for bigger problems the space requirement and the time requirement is not that high for the developed heuristic.
- When the number of stacks is increased, the time required is reduced as wella the number of steps are decreased.
- Both the heuristics that I developed are admissible as the heuristic did not over estimate the path cost.
- When number of stacks were 3 and the number of blocks were 16, it took the most frontier size, as there were inadequate number of stacks to move to. That's why the number of failures are increased in that space.
- The main idea on this heuristic development was try to find a "score" for each state analyzing successful solution path
- The second heuristic did not work as expected, but the third one combatted the issues the second heuristics dealt with.

- One of the largest problems that my algorithm was able to solve was the memory and time requirements. It took very less memory (biggest queue consisted on 9000 nodes), and took 2 seconds.
- The biggest problem that I was able to solve was 9 stacks and 23 blocks, taking nearly 4000 iterations an 1 minute time.
- In smaller problems, my queue grew twice the number of iterations but for bigger problems it was thrice or even 5 times.
- Increasing the number of stacks make the problems easier as the blocks had more stacks to go to, but the branching factor grew and as a consequence, the queue grew fast.
- My program generally find optimal solutions, but in bigger problems, I was not able to verify if the solution obtained was optimal or sub-optimal.
- Although the heuristic overestimates the actual path cost, its still performs better.
- Further, heuristics can be improved by integrating some more dependency factors in the calculation of the heuristics function. This would ensure that the next best block has minimum number of blocks over it, for it to quickly move over the current highest block in the first stack.

Sample optimum answer to the problem stated in the assignment (My best heuristic):

```
Initial State:
1 | ['A', 'D']
2 | ['B', 'E']
3 | ['C']

Iter= 1  Queue= 0  Depth= 0
Iter= 2  Queue= 5  Depth= 1
Iter= 3  Queue= 6  Depth= 2
Iter= 4  Queue= 9  Depth= 3
Iter= 5  Queue= 12  Depth= 4
Iter= 6  Queue= 15  Depth= 5
Iter= 7  Queue= 16  Depth= 6
Iter= 8  Queue= 19  Depth= 7
Iter= 9  Queue= 22  Depth= 8
Iter= 10  Queue= 23  Depth= 9
Iter= 11  Queue= 24  Depth= 10

GOAL REACHED, SUCCESS!
Number of iterations are :11
Frontier size is :24
Run time (in seconds) :0.314999818802
Depth of goal state is 10

Initial State:
1 | ['D']
2 | ['C', 'A']
```

```
3 | ['B', 'E']

Solution path is:

Next move:
1 | []
2 | ['C', 'A']
3 | ['B', 'E', 'D']

Next move:
1 | ['A']
2 | ['C']
3 | ['B', 'E', 'D']

Next move:
1 | ['A']
2 | ['C', 'D']
3 | ['B', 'E']

Next move:
1 | ['A']
2 | ['C', 'D', 'E']
3 | ['B']

Next move:
1 | ['A', 'B']
2 | ['C', 'D', 'E']
3 | []

Next move:
1 | ['A', 'B']
2 | ['C', 'D']
3 | ['E']

Next move:
1 | ['A', 'B']
2 | ['C']
3 | ['E', 'D']

Next move:
1 | ['A', 'B', 'C']
2 | []
3 | ['E', 'D']

Next move:
1 | ['A', 'B', 'C', 'D']
2 | []
3 | ['E']

Next move:
1 | ['A', 'B', 'C', 'D', 'E']
2 | []
3 | []
```