Resolução de problemas através de busca

Projeto 1 de CTC-17/2019

Prof. Paulo André Castro

Eduardo Alarcon[1], Victor da Rocha Sales[1]

[1]Aluno do quarto ano de Graduação em Engenharia da Computação do Instituto Tecnológico de Aeronáutica (ITA).

E-mail: eduambarbosa@gmail.com
salesrvictor@gmail.com

Data de entrega: 02/09/2019

I. INTRODUÇÃO

O primeiro projeto do laboratório de CTC-17 consiste de dois exercícios de programação de resolução de problemas através de busca usando os algoritmos greedy e A* para cada um deles. No primeiro exercício busca-se o menor caminho entre duas cidades obtendo a lista das cidades e a distância do início ao fim. Já o segundo exercício, cria-se um agente capaz de resolver o problema dos blocos deslizantes para nove blocos.

II. OBJETIVOS

O objetivo principal do primeiro projeto é exercitar e fixar os conhecimentos adquiridos sobre resolução de provas através de busca. Além disso, pode-se considerar os seguintes objetivos:

- Análise dos tipos de ambientes e busca.
- Utilização e comparação de estratégias de busca informada.
- Verificação da importância da consistência da heurística.

Para a elaboração dos exercícios, foi utilizado a linguagem Python.

III. DESCRIÇÃO

Para ambos os exercícios foi utilizado uma estrutura de heap para a ordenação dos nós. Os exercícios foram desenvolvidos de forma modular para compartilharem os algoritmos de busca, bem como a estrutura dos nós. Dessa forma, cada exercício é composto por uma classe principal, a qual contém as funções de cálculo de custo e de tomada de ações.

1. Exercício 1: menor caminho entre as cidades Alice Springs e Yulara da Australia

O mapa com as especificações das cidades (ID, nome, latitude, longitude, estado/território, população) está no arquivo “australia.csv”. A distância em linha reta entre as cidades pode ser calculada a partir das coordenadas cartesianas (latitude, longitude). A conexão entre as cidades é feita da seguinte forma: uma cidade com ID x se conecta com as cidades $x + 2$ e $x - 1$, se $x > 1$ e x é par; se x é ímpar e $x > 2$, esta cidade x se conecta com as cidades $x - 2$ e $x + 1$. Neste caso a distância pela estrada é 10% maior que a distância em linha reta.

Tanto para o algoritmo greedy quanto para o A* utilizou-se como heurística uma aproximação da distância até a cidade final como a distância em linha reta entre as duas cidades. Já para o algoritmo A*, o custo final considerado é a soma da heurística com o custo da distância a partir do nó inicial.

2. Exercício 2: agente capaz de resolver o problema dos blocos deslizantes para 9 blocos

O problema consiste de um tabuleiro de tamanho 9x9 com os blocos enumerados e com a última peça faltante. Com o espaço da peça faltante surgem as possibilidades de movimentação entre peças, movendo peças para o local vazio. O estado inicial é aleatório e o objetivo é alcançar o estado final ordenado do tabuleiro em que cada linha contém os valores crescentes da esquerda pra direita, começando pela linha superior e com o último bloco vazio.

Tanto para o algoritmo greedy quanto para o A* a heurística utilizada para aproximar o número de passos necessários para resolver o problema foi a soma, para cada bloco, da distância Manhattan entre onde este bloco estava e onde ele deveria estar, inclusive para o espaço vazio. Já para o algoritmo A*, o custo final considerado é a soma da heurística com o custo da distância a partir do nó inicial.

IV. RESULTADOS OBTIDOS

1. Exercício 1: menor caminho entre as cidades Alice Springs e Yulara da Australia

A comparação entre os resultados obtidos pelo algoritmo greedy e pelo algoritmo A* podem ser observadas na Tabela 1 a seguir.

Tabela 1. Profundidade e custo para o problema da Austrália para os algoritmos greedy e A*

Algoritmo	Profundidade	Custo
Greedy	151	2447.38
A*	163	2177.15

Para simplificar a visualização vamos observar a lista de cidades como uma lista das IDs, a lista com o nome das cidades está disponibilizada no arquivo “Exercicio_1.txt”.

- Solução do algoritmo Greedy:

[5, 6, 8, 7, 9, 10, 12, 14, 16, 18, 17, 19, 21, 22, 24, 26, 25, 27, 29, 30, 32, 31, 33, 35, 37, 38, 40, 42, 41, 43, 45, 47, 48, 50, 49, 51, 53, 55, 56, 58, 57, 59, 61, 63, 65, 66, 68, 70, 72, 74, 73, 75, 77, 79, 80, 82, 81, 83, 84, 86, 88, 87, 89, 90, 92, 94, 96, 95, 97, 99, 100, 102, 104, 106, 108, 110, 112, 114, 113, 115, 117, 118, 120, 122, 121, 123, 125, 127, 129, 130, 132, 134, 133, 135, 137, 138, 140, 142, 141, 143, 145, 147, 149, 151, 153, 155, 156, 158, 157, 159, 161, 162, 164, 166, 168, 167, 169, 171, 172, 174, 176, 175, 177, 178, 180, 182, 184, 186, 185, 187, 189, 191, 192, 194, 196, 195, 197, 199, 200, 202, 204, 203, 205, 207, 209, 211, 212, 214, 213, 215, 217, 219]

- Solução do algoritmo A*:

[5, 6, 8, 7, 9, 10, 12, 11, 13, 14, 16, 18, 17, 19, 21, 22, 24, 26, 25, 27, 29, 30, 32, 34, 33, 35, 37, 39, 40, 42, 41, 43, 45, 47, 48, 50, 49, 51, 53, 55, 56, 58, 57, 59, 61, 63, 65, 66, 68, 70, 72, 71, 73, 74, 76, 75, 77, 79, 80, 82, 81, 83, 85, 86, 88, 87, 89, 90, 92, 91, 93, 95, 96, 98, 97, 99, 100, 102,

104, 103, 105, 106, 108, 110, 112, 111, 113, 114, 116, 115, 117, 118, 120, 119, 121, 123, 124, 126, 128, 130, 129, 131, 133, 134, 136, 138, 137, 139, 140, 142, 141, 143, 144, 146, 148, 147, 149, 151, 153, 155, 157, 159, 161, 162, 164, 166, 168, 170, 172, 174, 176, 175, 177, 179, 180, 182, 184, 186, 185, 187, 189, 190, 192, 191, 193, 194, 196, 195, 197, 199, 201, 203, 205, 206, 208, 210, 209, 211, 212, 214, 213, 215, 217, 219]

2. Exercício 2: agente capaz de resolver o problema dos blocos deslizantes para 9 blocos

Para comparar os desempenhos do algoritmo greedy com o A* foram feitos três testes aleatórios de complexidades crescentes e dois testes propostos pelo roteiro. A representação é feita como greedy na esquerda e A* na direita, para poder compará-los de maneira prática.

O primeiro teste está representado a seguir.

========= USING GREEDY =========

Step: 1

State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	
64	65	66	67	68	69	70	71	63
73	74	75	76	77	78	79	80	72

Step: 2

State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	
73	74	75	76	77	78	79	80	72

Step: 3

State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	

========= USING A* =========

Step: 1

State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	
64	65	66	67	68	69	70	71	63
73	74	75	76	77	78	79	80	72

Step: 2

State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	
73	74	75	76	77	78	79	80	72

Step: 3

State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	

O segundo teste está representado a seguir.

========= USING GREEDY =========

Step: 1

State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52		53
55	56	57	58	59	60	61	62	54
64	65	66	67	68	69	70	72	63
73	74	75	76	77	78	79	71	80

Step: 2

State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	
55	56	57	58	59	60	61	62	54
64	65	66	67	68	69	70	72	63
73	74	75	76	77	78	79	71	80

Step: 3

State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	
64	65	66	67	68	69	70	72	63
73	74	75	76	77	78	79	71	80

Step: 4

State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	72	
73	74	75	76	77	78	79	71	80

========= USING A* =========

Step: 1

State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52		53
55	56	57	58	59	60	61	62	54
64	65	66	67	68	69	70	72	63
73	74	75	76	77	78	79	71	80

Step: 2

State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	
55	56	57	58	59	60	61	62	54
64	65	66	67	68	69	70	72	63
73	74	75	76	77	78	79	71	80

Step: 3

State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	
64	65	66	67	68	69	70	72	63
73	74	75	76	77	78	79	71	80

Step: 4

State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	72	
73	74	75	76	77	78	79	71	80

Step: 5
State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	72	80
73	74	75	76	77	78	79	71	

Step: 5
State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70		72
73	74	75	76	77	78	79	71	80

Step: 6
State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	72	80
73	74	75	76	77	78	79		71

Step: 6
State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79		80

Step: 7
State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70		80
73	74	75	76	77	78	79	72	71

Step: 7
State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	

Step: 8
State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	80	
73	74	75	76	77	78	79	72	71

Step: 9
State:

1	2	3	4	5	6	7	8	9

10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	80	71
73	74	75	76	77	78	79	72	

Step: 10
State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	80	71
73	74	75	76	77	78	79		72

Step: 11
State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70		71
73	74	75	76	77	78	79	80	72

Step: 12
State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	
73	74	75	76	77	78	79	80	72

Step: 13
State:

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36

37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	

O terceiro teste, e os dois testes propostos pelo roteiro são muito grandes. Dessa forma, podem ser observados nos arquivos “Exercicio_2_T3.txt”, “Exercicio_2_R1.txt” e “Exercicio_2_R2.txt”. Seus resultados estão compilados na Tabela 2 a seguir.

Tabela 2. Quantidade de passos para o algoritmo greedy e A* para o terceiro teste, primeiro do roteiro e segundo do roteiro.

Algoritmo	Terceiro teste	Primeiro do roteiro	Segundo do roteiro
Greedy	227	228	maximum recursion depth
A*	11	10	maximum recursion depth

V. CONCLUSÃO

Para o primeiro exercício, apesar da profundidade do algoritmo A* acabar sendo maior que a profundidade do algoritmo greedy, o custo do A* é menor. Já para o segundo exercício, percebe-se uma menor profundidade para o algoritmo A*, sendo ela justamente o tamanho da solução do problema. Dessa forma, pode-se perceber que a adição da distância até o nó inicial contribui com o algoritmo ao evitar expandir caminhos que já ficaram caros, fato ignorado pelo algoritmo greedy (guloso) em ambos os exercícios.