

# CS221 Programming Assignment 1 README

Anand Madhavan - 05401919 <[manand@stanford.edu](mailto:manand@stanford.edu)>

Elizabeth Lingg - 05215856 <[elingg@stanford.edu](mailto:elingg@stanford.edu)>

Alec Go - 005516471 <[alecmgo@stanford.edu](mailto:alecmgo@stanford.edu)>

---

## Results

The following table shows the optimality of each algorithm:

Algorithm	Optimal
Best-first search	No
Uniform cost search	Yes
A*	Yes
A* with an inadmissible heuristic	No

Run-time performance can be estimated by the number of nodes checked before finding a solution. The two algorithms that were not optimal (best first search and A\* with an inadmissible heuristic) tended to find a solution faster than the two algorithms that were optimal (uniform cost search and A\*). This makes sense because there are many solutions, but only one optimal solution, so it will take longer to find an optimal solution. When comparing the optimal algorithms, A\* found a solution faster than uniform cost search.

Introducing an inadmissible heuristic to A\* caused it to sometimes miss the optimal solution. For the specific inadmissible heuristic we chose ( $100 \cdot h$ ), the algorithm found a solution faster (in general).

The following tables shows the results for 4 test runs.

### Path length of found solution

	World 1, 26 Points	World 1, 49 Points	World 2, 49 Points	World 2, 60 Points
<b>Best-first search</b>	12.754813	5.905653	5.210925	5.210925
<b>Uniform cost search</b>	8.514715	5.905653	5.210925	5.210925
<b>A*</b>	8.514715	5.905653	5.210925	5.210925
<b>A* with an inadmissible heuristic</b>	10.760715	5.905653	5.210925	5.210925

### Number of nodes checked before finding solution

	World 1, 26 Points	World 1, 49 Points	World 2, 49 Points	World 2, 60 Points
<b>Best-first search</b>	18	8	6	6
<b>Uniform cost search</b>	19	39	35	44
<b>A*</b>	19	22	9	9
<b>A* with an inadmissible heuristic</b>	18	9	6	6

### Time elapsed (ms)

	World 1, 26 Points	World 1, 49 Points	World 2, 49 Points	World 2, 60 Points
<b>Best-first search</b>	260	830	1040	1570
<b>Uniform cost search</b>	270	840	1060	1590
<b>A*</b>	260	840	1050	1560
<b>A* with an inadmissible heuristic</b>	260	830	1050	1560

---

## Code Overview

We implemented best first search, uniform cost search, A-star, and A-star with an inadmissible heuristic. These algorithms all shared a common algorithm that uses a base class 'Algo' (in agent.hh) that provides the heuristic and path costs. For each algorithm, we extended the 'Algo' class (again in agent.hh) and implemented corresponding methods for calculating heuristic and path costs.

The common algorithm works as follows:

- We include the 'goal' and 'start' nodes into the list of points during 'init' of the Agent.
- We create a roadmap (for efficiency) that maps a given point to its visible neighbors. This was made efficient by updating the map for mutual neighbors, when two points were detected as visible neighbors. The roadmap in turn results in better efficiency while traversing the graph.

- We also maintain a set of expanded point-indexes, so we can eliminate duplicate expansions when we encounter them in the graph traversal.
  - Each QueueNode in the priority queue was also extended to have the following information:
    - order-number: which is the order in which it was inserted into the priority queue (this was done for breaking ties based on the FIFO policy suggested in the handout)
    - point-index: the state consists of the path to the last point in the path. This point-index corresponds to the last point in the path, as ordered in the 'points' vector of the agent class. This was introduced for efficiency for indexing into the roadmap structure for example.
  - We now begin our traversals. We add the start node to the priority queue.
  - We proceed to loop through the priority queue (the algorithm now is as suggested in the lecture notes), popping each element and expanding out nodes that are not already expanded. The expansion is done using the neighbors suggested in the roadmap. Nodes when expanded are added to the priority queue using the costs computed by the specific 'Algo' class passed in.
  - We also modified the priorityqueue class's () operator to use the order-number in breaking ties. The lower the order-number, the higher the priority (based on the FIFO policy suggested).
- 

## Output

```
./rpplan -text -file world1.config -seed 10 -points 26
```

```
Seed: 10
```

```
2
```

```
WorldFile: world1.config
```

```
generating points
```

```
done generating points
```

```
Performing best first search
```

```
Found path after checking 18 nodes
```

```
Solution:
```

```
1   Start   (0.000000, 0.000000, 0.000000, 0.000000)
2           (0.463714, -0.154194, -0.211811, -1.317227)
3           (1.371192, -2.126400, -0.925601, 2.781327)
4           (0.471146, -1.688551, -0.790313, -1.793997)
5   Finish  (-0.589000, -0.589000, -0.589000, -0.589000)
```

```
Path length: 12.754813
```

```
Uniform cost search
```

```
Found path after checking 19 nodes
```

```
Solution:
```

```
1   Start   (0.000000, 0.000000, 0.000000, 0.000000)
2           (1.360652, -1.685585, -0.273146, 1.631041)
3           (1.371192, -2.126400, -0.925601, 2.781327)
4           (-0.109130, -1.355227, -0.714068, 1.502653)
5   Finish  (-0.589000, -0.589000, -0.589000, -0.589000)
```

Path length: 8.514715

Performing A\*

Found path after checking 19 nodes

Solution:

1	Start	(0.000000, 0.000000, 0.000000, 0.000000)
2		(1.360652, -1.685585, -0.273146, 1.631041)
3		(1.371192, -2.126400, -0.925601, 2.781327)
4		(-0.109130, -1.355227, -0.714068, 1.502653)
5	Finish	(-0.589000, -0.589000, -0.589000, -0.589000)

Path length: 8.514715

Performing inadmissible A\*

Found path after checking 18 nodes

Solution:

1	Start	(0.000000, 0.000000, 0.000000, 0.000000)
2		(1.360652, -1.685585, -0.273146, 1.631041)
3		(1.371192, -2.126400, -0.925601, 2.781327)
4		(0.471146, -1.688551, -0.790313, -1.793997)
5	Finish	(-0.589000, -0.589000, -0.589000, -0.589000)

Path length: 10.760715

---

./rpplan -text -file world1.config -seed 10 -points 49

Seed: 10

2

WorldFile: world1.config

generating points

done generating points

Performing best first search

Found path after checking 8 nodes

Solution:

1	Start	(0.000000, 0.000000, 0.000000, 0.000000)
2		(1.148721, -0.174491, -0.780260, -0.966965)
3		(0.449850, -1.280280, -1.835320, -1.976533)
4	Finish	(-0.589000, -0.589000, -0.589000, -0.589000)

Path length: 5.905653

Uniform cost search

Found path after checking 39 nodes

Solution:

1	Start	(0.000000, 0.000000, 0.000000, 0.000000)
2		(1.148721, -0.174491, -0.780260, -0.966965)
3		(0.449850, -1.280280, -1.835320, -1.976533)
4	Finish	(-0.589000, -0.589000, -0.589000, -0.589000)

Path length: 5.905653

Performing A\*

Found path after checking 22 nodes

Solution:

```
1    Start  (0.000000, 0.000000, 0.000000, 0.000000)
2          (1.148721, -0.174491, -0.780260, -0.966965)
3          (0.449850, -1.280280, -1.835320, -1.976533)
4    Finish  (-0.589000, -0.589000, -0.589000, -0.589000)
```

Path length: 5.905653

Performing inadmissible A\*

Found path after checking 9 nodes

Solution:

```
1    Start  (0.000000, 0.000000, 0.000000, 0.000000)
2          (1.148721, -0.174491, -0.780260, -0.966965)
3          (0.449850, -1.280280, -1.835320, -1.976533)
4    Finish  (-0.589000, -0.589000, -0.589000, -0.589000)
```

Path length: 5.905653

---

```
./rpplan -text -file world2.config -seed 10 -points 49
```

Seed: 10

2

WorldFile: world2.config

generating points

done generating points

Performing best first search

Found path after checking 6 nodes

Solution:

```
1    Start  (0.000000, 0.000000, 0.000000, 0.000000, 0.000000)
2          (1.196058, -1.987440, 0.581121, -1.217120, 0.859224)
3    Finish  (-0.786000, -0.786000, 0.786000, -0.786000, 0.500000)
```

Path length: 5.210925

Uniform cost search

Found path after checking 35 nodes

Solution:

```
1    Start  (0.000000, 0.000000, 0.000000, 0.000000, 0.000000)
2          (1.196058, -1.987440, 0.581121, -1.217120, 0.859224)
3    Finish  (-0.786000, -0.786000, 0.786000, -0.786000, 0.500000)
```

Path length: 5.210925

Performing A\*

Found path after checking 9 nodes

Solution:

```
1    Start  (0.000000, 0.000000, 0.000000, 0.000000, 0.000000)
2          (1.196058, -1.987440, 0.581121, -1.217120, 0.859224)
3    Finish  (-0.786000, -0.786000, 0.786000, -0.786000, 0.500000)
```

Path length: 5.210925

Performing inadmissible A\*

Found path after checking 6 nodes

Solution:

```
1    Start  (0.000000, 0.000000, 0.000000, 0.000000, 0.000000)
2              (1.196058, -1.987440, 0.581121, -1.217120, 0.859224)
3    Finish  (-0.786000, -0.786000, 0.786000, -0.786000, 0.500000)
```

Path length: 5.210925

---

./rpplan -text -file world2.config -seed 10 -points 60

Seed: 10

2

WorldFile: world2.config

generating points

done generating points

Performing best first search

Found path after checking 6 nodes

Solution:

```
1    Start  (0.000000, 0.000000, 0.000000, 0.000000, 0.000000)
2              (1.196058, -1.987440, 0.581121, -1.217120, 0.859224)
3    Finish  (-0.786000, -0.786000, 0.786000, -0.786000, 0.500000)
```

Path length: 5.210925

Uniform cost search

Found path after checking 44 nodes

Solution:

```
1    Start  (0.000000, 0.000000, 0.000000, 0.000000, 0.000000)
2              (1.196058, -1.987440, 0.581121, -1.217120, 0.859224)
3    Finish  (-0.786000, -0.786000, 0.786000, -0.786000, 0.500000)
```

Path length: 5.210925

Performing A\*

Found path after checking 9 nodes

Solution:

```
1    Start  (0.000000, 0.000000, 0.000000, 0.000000, 0.000000)
2              (1.196058, -1.987440, 0.581121, -1.217120, 0.859224)
3    Finish  (-0.786000, -0.786000, 0.786000, -0.786000, 0.500000)
```

Path length: 5.210925

Performing inadmissible A\*

Found path after checking 6 nodes

Solution:

```
1    Start  (0.000000, 0.000000, 0.000000, 0.000000, 0.000000)
2              (1.196058, -1.987440, 0.581121, -1.217120, 0.859224)
3    Finish  (-0.786000, -0.786000, 0.786000, -0.786000, 0.500000)
```

Path length: 5.210925