

sim_astar

October 17, 2019

1 A* Motion Planning

```
[1]: # The autoreload extension will automatically load in new code as you edit ↵
      ↪files,
      # so you don't need to restart the kernel every time
      %load_ext autoreload
      %autoreload 2
      import numpy as np
      import matplotlib.pyplot as plt
      from P1_astar import DetOccupancyGrid2D, AStar
      from utils import generate_planning_problem
```

1.1 Simple Environment

1.1.1 Workspace

(Try changing this and see what happens)

```
[2]: width = 10
      height = 10
      obstacles = [((6,7),(8,8)),((2,2),(4,3)),((2,5),(4,7)),((6,3),(8,5))]
```

1.1.2 Starting and final positions

(Try changing these and see what happens)

```
[3]: x_init = (1, 1)
      x_goal = (9, 9)
```

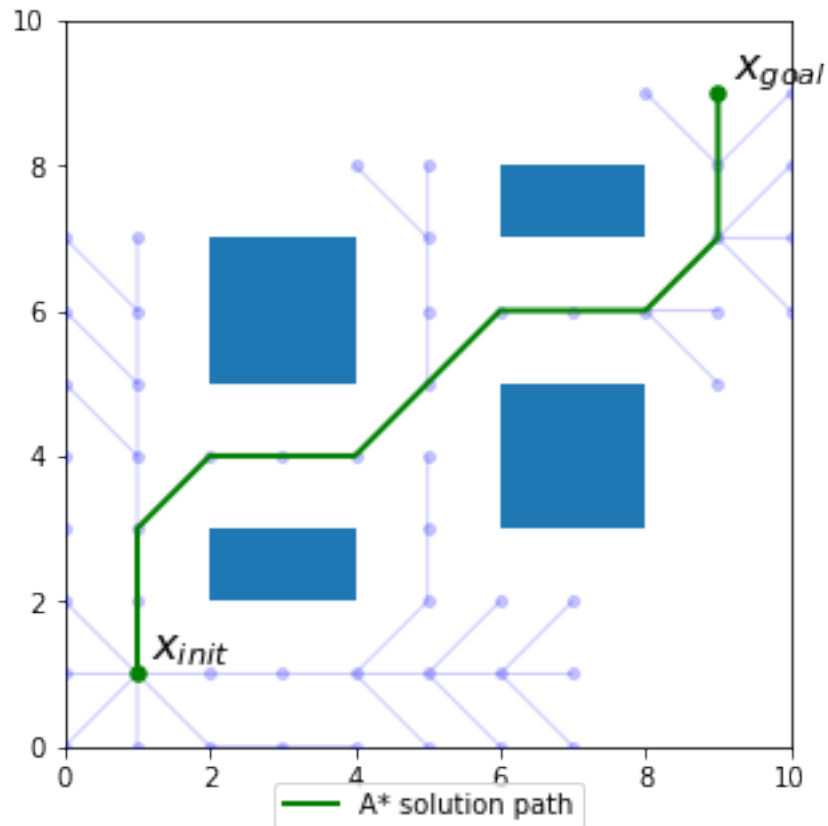
1.1.3 Run A* planning

```
[4]: astar = AStar((0, 0), (width, height), x_init, x_goal, occupancy)
      if not astar.solve():
          print "No path found"
      else:
          plt.rcParams['figure.figsize'] = [5, 5]
          astar.plot_path()
```

```
astar.plot_tree()
```

/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/site-packages/matplotlib/cbook/deprecation.py:107: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instance. In a future version, a new instance will always be created and returned. Meanwhile, this warning can be suppressed, and the future behavior ensured, by passing a unique label to each axes instance.

```
warnings.warn(message, mplDeprecation, stacklevel=1)
```



1.2 Random Cluttered Environment

1.2.1 Generate workspace, start and goal positions

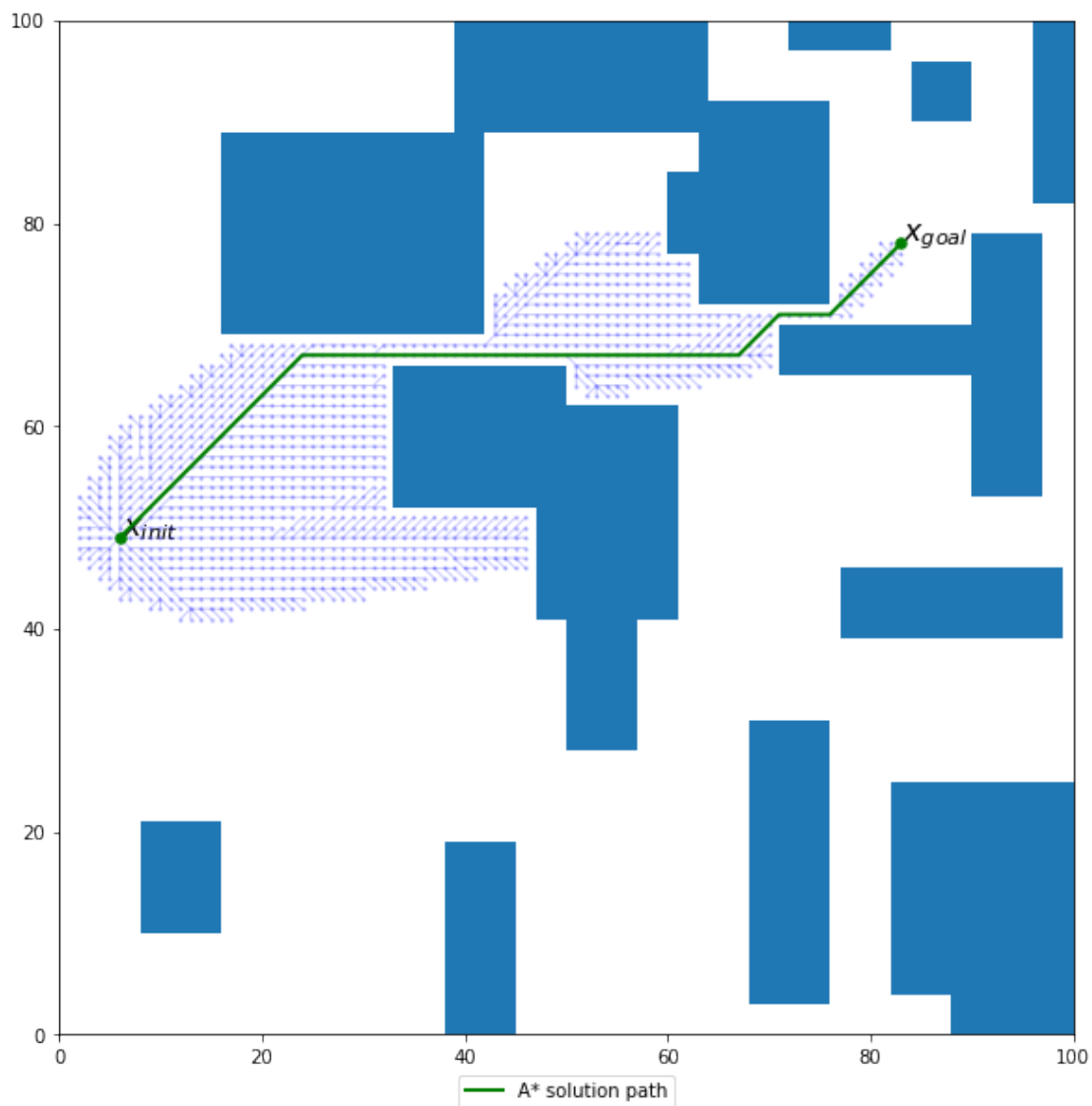
(Try changing these and see what happens)

```
[5]: width = 100
height = 100
num_obs = 25
min_size = 5
max_size = 30
```

```
occupancy, x_init, x_goal = generate_planning_problem(width, height, num_obs,
↳ min_size, max_size)
```

1.2.2 Run A* planning

```
[6]: astar = AStar((0, 0), (width, height), x_init, x_goal, occupancy)
if not astar.solve():
    print "No path found"
else:
    plt.rcParams['figure.figsize'] = [10, 10]
    astar.plot_path()
    astar.plot_tree(point_size=2)
```



[]: