robot_navigation

April 9, 2024

[121]:
```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import sys
```

[122]:
```
sys.path.append("../")
```

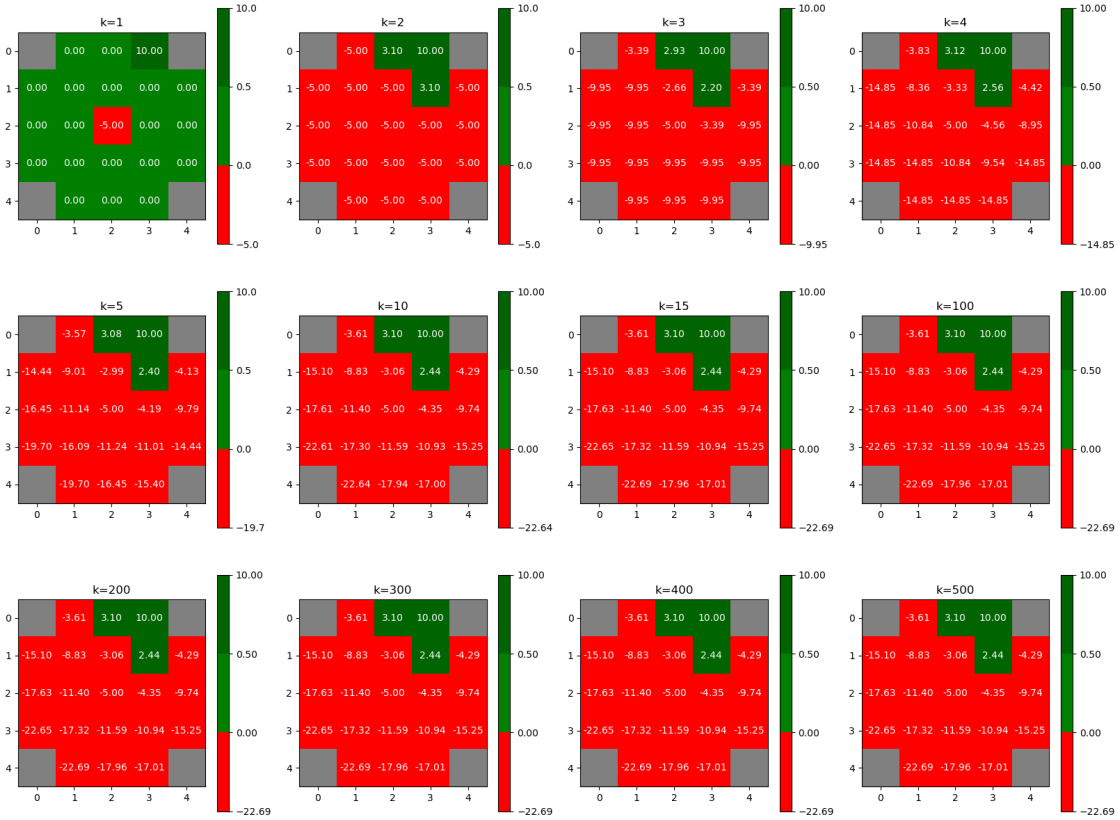
[123]:
```
from robot_navigation.generator import MazeGenerator
from robot_navigation.model import MazeConfig, PolicyConfig, Reward
from robot_navigation.optimizer import MazeOptimizer
from robot_navigation.viz import MazeVisualizer
```

1 T1

1.1 R1, R2 and r (i.e., 10, -5, -5), and P1

[124]:
```
R1 = 10
R2 = -5
r = living_reward = -5

destination_reward = Reward(row=0, column=3, value=R1)
hazard_reward = Reward(row=2, column=2, value=R2)
obstacle_1 = Reward(row=0, column=0, value=np.nan)
obstacle_2 = Reward(row=0, column=4, value=np.nan)
obstacle_3 = Reward(row=4, column=0, value=np.nan)
obstacle_4 = Reward(row=4, column=4, value=np.nan)
rewards = [destination_reward, hazard_reward, obstacle_1, obstacle_2,␣
 ↪obstacle_3, obstacle_4]
maze_config = MazeConfig(rows=5, columns=5, rewards=rewards)

maze = MazeGenerator.generate_maze(maze_config)

intended_direction=0.90
unintended_direction=0.10

k = [1, 2, 3, 4, 5, 10, 15, 100, 200, 300, 400, 500]
```

[125]:
```
MazeVisualizer.visualize_map_iterations(maze_config, living_reward,␣
↪intended_direction, unintended_direction, k)
```



[126]:
```
policy_config = PolicyConfig(maze=maze,
                             rewards=rewards,
                             iterations=500,
                             living_reward=living_reward,
                             intended_direction=0.80,
                             unintended_direction=0.10)
```

[127]:
```
result_maze = MazeOptimizer.optimize_maze(policy_config)
result_maze
```

[127]:
```
Maze(map=array([[         nan,  -4.8591731 ,   1.99201605,  10.        ,
                  nan],
        [-14.59040416,  -9.51242745,  -4.30305211,   1.32000656,
          -5.47497497],
        [-15.87178161, -10.85553741,  -5.        ,  -5.43224768,
         -10.36337717],
        [-19.36666464, -15.54961806, -11.01048698, -11.23433598,
         -14.8048019 ],
```

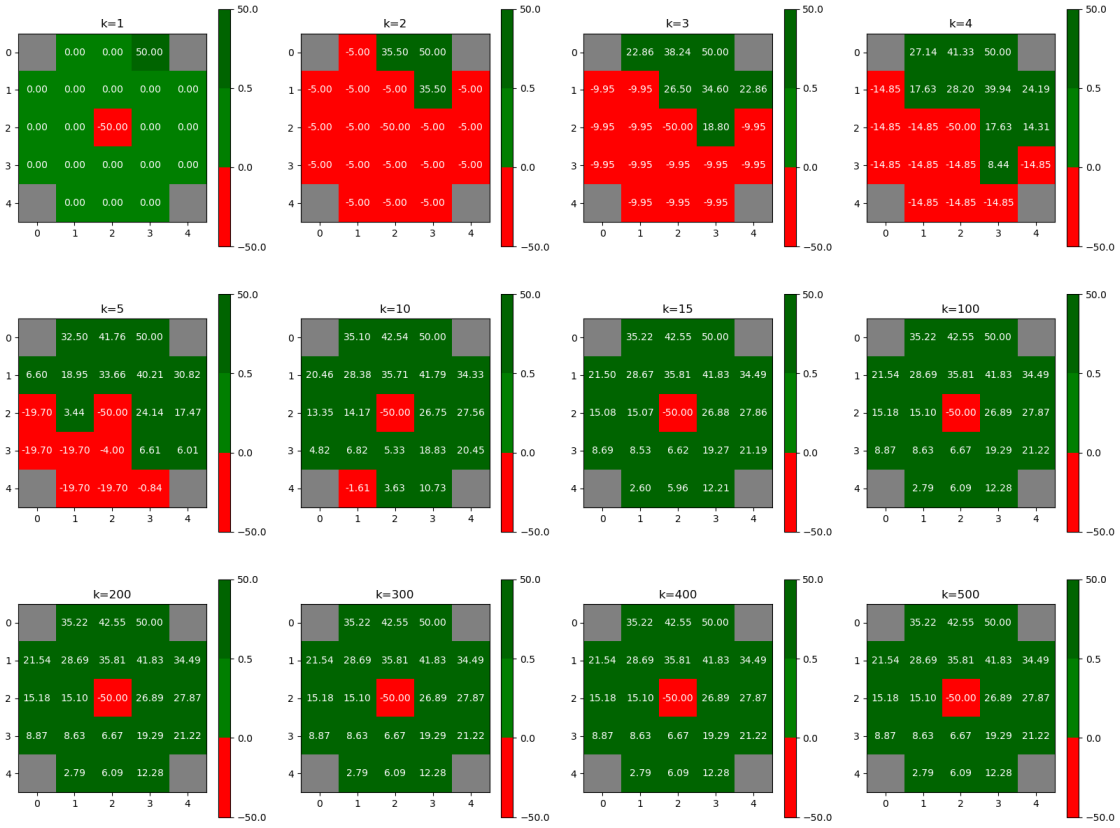
```
[         nan, -19.3902069 , -16.11005097, -15.97607444,
          nan]]))
```

2 T3

2.1 R1, R2 and r (i.e., 50, -50, -5), and P2

[128]:
```
R1 = 50
R2 = -50
r = living_reward = -5

destination_reward = Reward(row=0, column=3, value=R1)
hazard_reward = Reward(row=2, column=2, value=R2)

rewards = [destination_reward, hazard_reward, obstacle_1, obstacle_2,␣
 ↪obstacle_3, obstacle_4]
maze_config = MazeConfig(rows=5, columns=5, rewards=rewards)
```

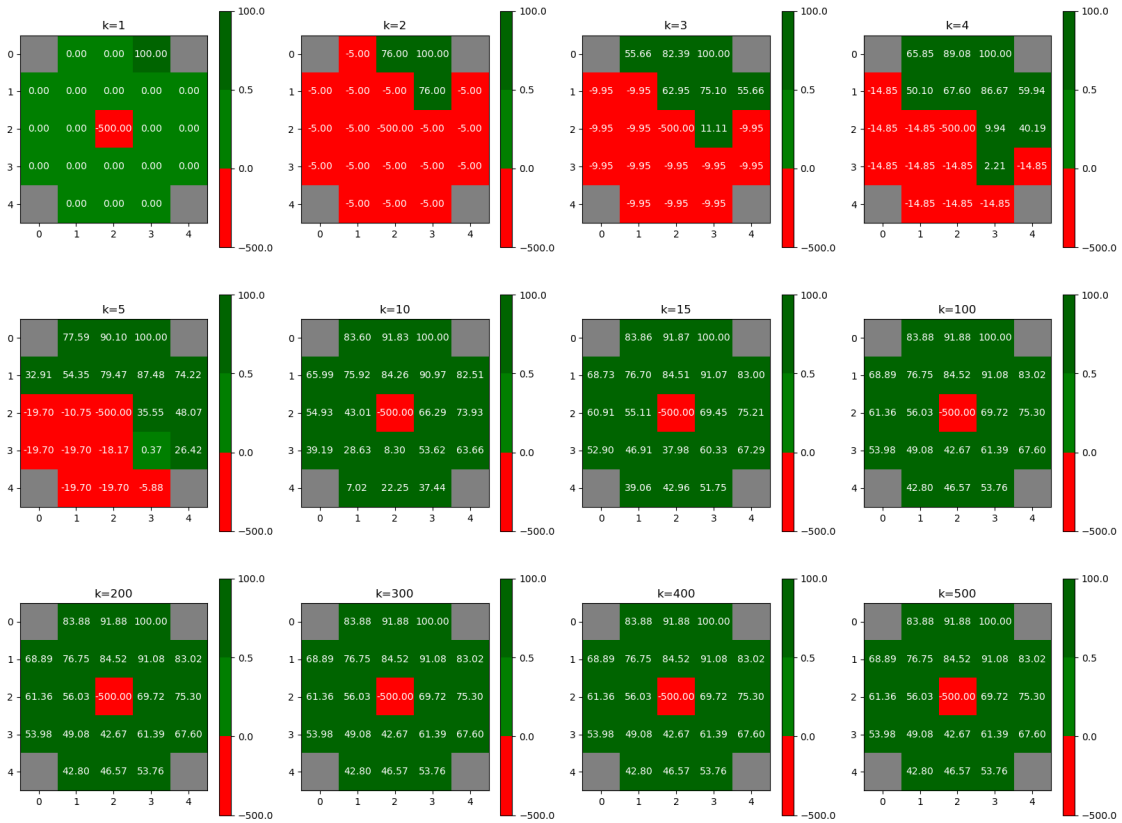
[129]:
```
MazeVisualizer.visualize_map_iterations(maze_config, living_reward,␣
 ↪intended_direction, unintended_direction, k)
```



2.2 R1, R2 and r (i.e., 100, -500, -5), and P3

[130]:
```
R1 = 100
R2 = -500
r = living_reward = -5

destination_reward = Reward(row=0, column=3, value=R1)
hazard_reward = Reward(row=2, column=2, value=R2)

rewards = [destination_reward, hazard_reward, obstacle_1, obstacle_2,␣
 ↪obstacle_3, obstacle_4]
maze_config = MazeConfig(rows=5, columns=5, rewards=rewards)
```

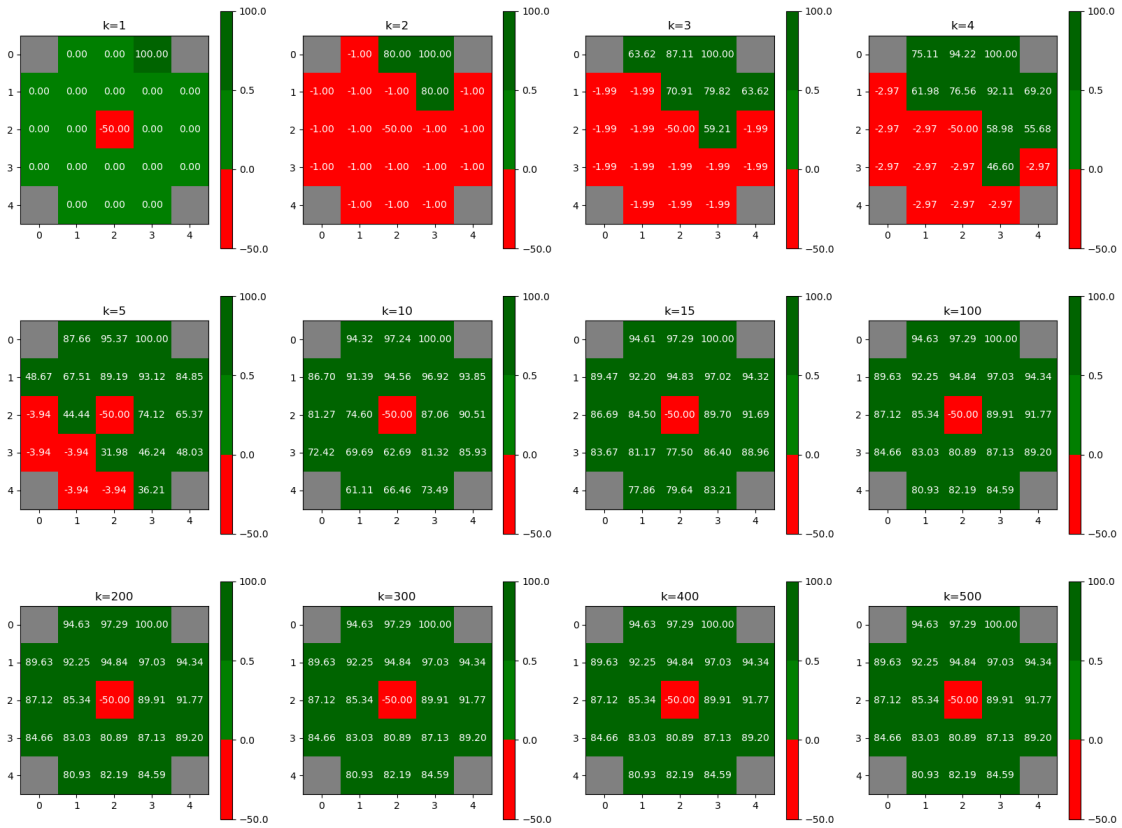
[131]:
```
MazeVisualizer.visualize_map_iterations(maze_config, living_reward,␣
 ↪intended_direction, unintended_direction, k)
```



2.3 R1, R2 and r (i.e., 100, -50, -1), and P4.a

[132]:
```
R1 = 100
R2 = -50
r = living_reward = -1

destination_reward = Reward(row=0, column=3, value=R1)
hazard_reward = Reward(row=2, column=2, value=R2)

rewards = [destination_reward, hazard_reward, obstacle_1, obstacle_2,␣
 ↪obstacle_3, obstacle_4]
maze_config = MazeConfig(rows=5, columns=5, rewards=rewards)
```

[133]:
```
MazeVisualizer.visualize_map_iterations(maze_config, living_reward,␣
 ↪intended_direction, unintended_direction, k)
```



2.4 R1, R2 and r (i.e., 100, -50, -5), and P4.b

[134]:
```
R1 = 100
R2 = -50
r = living_reward = -5

destination_reward = Reward(row=0, column=3, value=R1)
hazard_reward = Reward(row=2, column=2, value=R2)

rewards = [destination_reward, hazard_reward, obstacle_1, obstacle_2,␣
 ↪obstacle_3, obstacle_4]
maze_config = MazeConfig(rows=5, columns=5, rewards=rewards)
```

[135]:
```
MazeVisualizer.visualize_map_iterations(maze_config, living_reward,␣
 ↪intended_direction, unintended_direction, k)
```