

Path Planning with Numpy Example

```
In [1]: from path_planning_numpy import *
```

Setting the Environment Parameters

```
In [2]: start=np.array([0,0])
end=np.array([10,10])
min_size=0.5
max_size=1
per_obs_coef=0.5
step=0.5
```

Creating Randomized Obstacles.

The first number is the number of obstacles created about the line connecting start and end. and the second number is the number of obstacles created in the environment.

```
In [3]: obstacles = create_random_obstacles(4,40,start,end,min_size,max_size,per_obs_coef)
print("The number of random obstacles created:",len(obstacles))
```

The number of random obstacles created: 44

Running the Path Planning Algorithms

Finds all the possible paths from start to end

```
In [4]: paths = finding_all_routes(start,end,step,obstacles)
```

All possible routes calculated. Total routes: 35

Finding the shortest path based on distance travelled

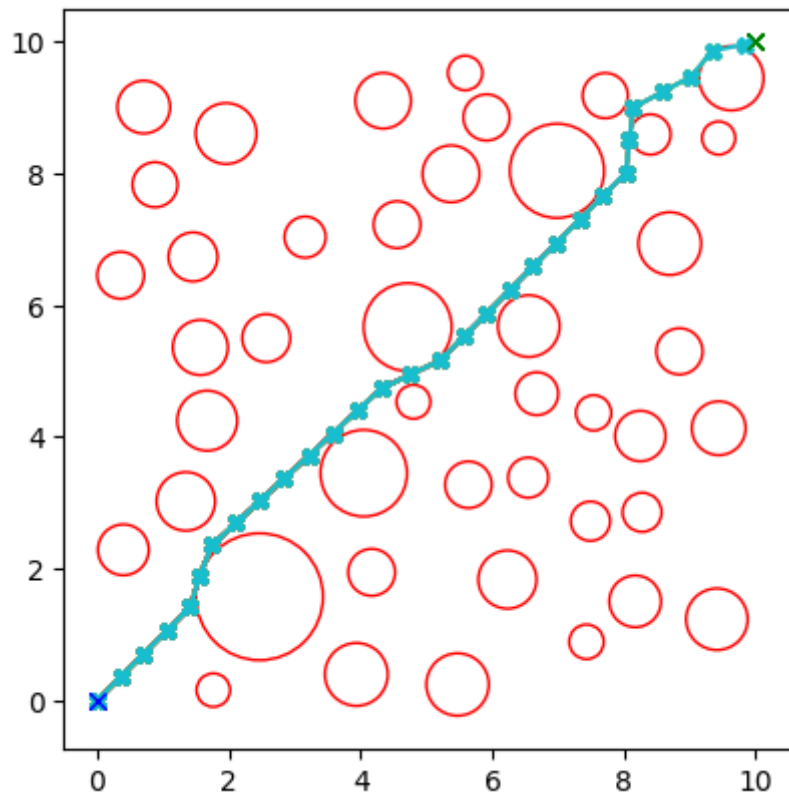
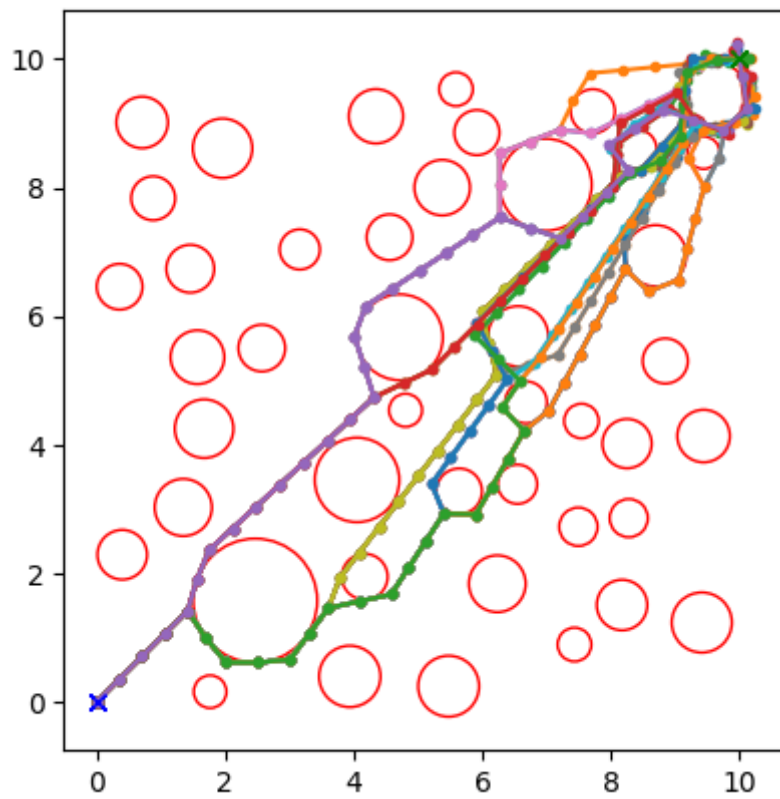
```
In [5]: shortest_p = shortest_path(paths)
print("The shortest path has a length of: {:.4f}".format(shortest_p.calculate_distance()))
```

The shortest path has a length of: 14.5000

Plotting the results

(all paths + optimal path)

```
In [6]: figure, axes = plt.subplots()
plot_path(paths,start,end,obstacles,axes)
figure2, axes2 = plt.subplots()
plot_path(shortest_p,start,end,obstacles,axes2)
plt.show()
```



In []: