

Problem:

This is an idealization of a problem where a robot has to navigate its way around obstacles. The goal is to find the shortest distance between two points on a plane that has convex polygonal obstacles. Figure 1 shows an example scene with twelve polygonal obstacles where the robot has to move from the point *start* to the point *end*.

We will assume that the shortest path from one polygon vertex to any other in the scene consists of straight-line segments joining some of the vertices of the polygon. Also we will assume that the start and the goal points may be considered polygons of size 0.

We will present the solutions for the scene specified below with 12 polygonal obstacles:

Polygon 1: ((220, 616), (220, 666), (251, 670), (272, 647))

Polygon 2: ((341, 655), (359, 667), (374, 651), (366, 577))

Polygon 3: ((311, 530), (311, 559), (339, 578), (361, 560), (361, 528), (336, 516))

Polygon 4: ((105, 628), (151, 670), (180, 629), (156, 577), (113, 587))

Polygon 5: ((118, 517), (245, 517), (245, 557), (118, 557))

Polygon 6: ((280, 583), (333, 583), (333, 665), (280, 665))

Polygon 7: ((252, 594), (290, 562), (264, 538))

Polygon 8: ((198, 635), (217, 574), (182, 574))

Polygon 9: ((190, 675), (210, 675), (210, 650), (190, 645))

Polygon 10: ((280, 540), (305, 550), (300, 510), (280, 510))

Polygon 11: ((230, 600), (250, 620), (240, 580))

Polygon 12: ((270, 680), (360, 695), (340, 675), (260, 666))

Start: (110 530)

End: (350 680)

Solution:

The state space is represented by the number of vertices for each polygon. Given the fact that the start and end point are also polygons of size zero then the complete number of vertices is: $48 + 2 = 50$. So the state space size is 50 points.

We will implement **Algorithm A*** (A Star) to find the shortest path from the start node to the end node using heuristic search. We will use the straight-line distance to the end node as a heuristic function.

We will define the necessary functions to implement the search problem. This should include a function that takes a vertex as input and returns the set of vertices that can be reached in a straight line from the given vertex.



Figure 1- Sample Scene

Code is included in Program_AI_1_cs. An executable file “Robot.exe” is also added. To use the program “Robot.exe”, please run the following from the command prompt in the same directory where the file exists:

Robot StartX StartY GoalX GoalY

The output will be similar to the following when running the following command (Test Case 3 provided for this exercise):

Directory\robot 370 550 110 660

They are:

False

Total Cost is : 308.861084278079

Path is:

110 660

151 670

198 635

220 616

250 620

280 583

339 578

361 560

370 550

The C# Source Code is also provided in Program_AI_1.cs files.