3.1.
when w =0, f(n)=2g(n), it’s uniform-cost search.
when w =1, f(n)=g(n)+h(n), it’s a A* search.
when w =2, f(n)=2h(n), it’s greedy best-first search.

To determine the values of w that makes the algorithm optimal, we can write the function as:
f(n) = (2-w)[g(n)+(w/(2-w))h(n)]
it behaves exactly like A* search with a heuristic (w/(2-w))h(n). It is optimal for w<=1, this is always less than h(n) and hence admissible, provided h(n) is admissible.

3.2
1.
The state space consists of all the (x,y) positions in the plane. A plane contains infinite number of points, the state space contains infinite number of points. So the number of states is also infinite. The paths are links between states. Therefore, the number of states and the number of paths to goal state are infinite.
2.
The shortest path from one polygon vertex to another point is a straight line. If there are obstacles in the path, then the shortest path is to wrap the path as close to a straight line around those obstacles as possible. Therefore the good state space contains the vertices of the polygon, the start point, the goal point.
3.
The heuristic function h(n)=straight line distance from vertex n to goal vertex
The successor function ACTIONS(vertex)
return a set of vectors contains the vertices that are reachable in a straight line, from the current vertex.
The RESULT function returns successor vertex such that the straight line distance from that vertex to goal is minimum.
function RESULT(v,ACTIONS(v))
min-straightlinedistance < = = INFINITY
successor < = = v
for each (x,y) in ACTIONS(v):
if x=v and min-straightlinedistance>h(y)
min-straightlinedistance = h(y)
successor = y
return successor