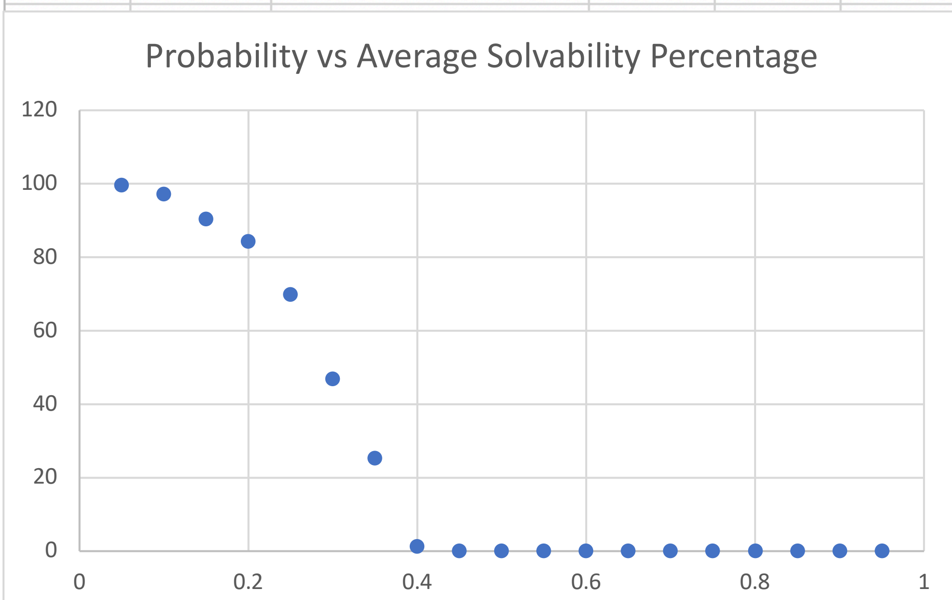
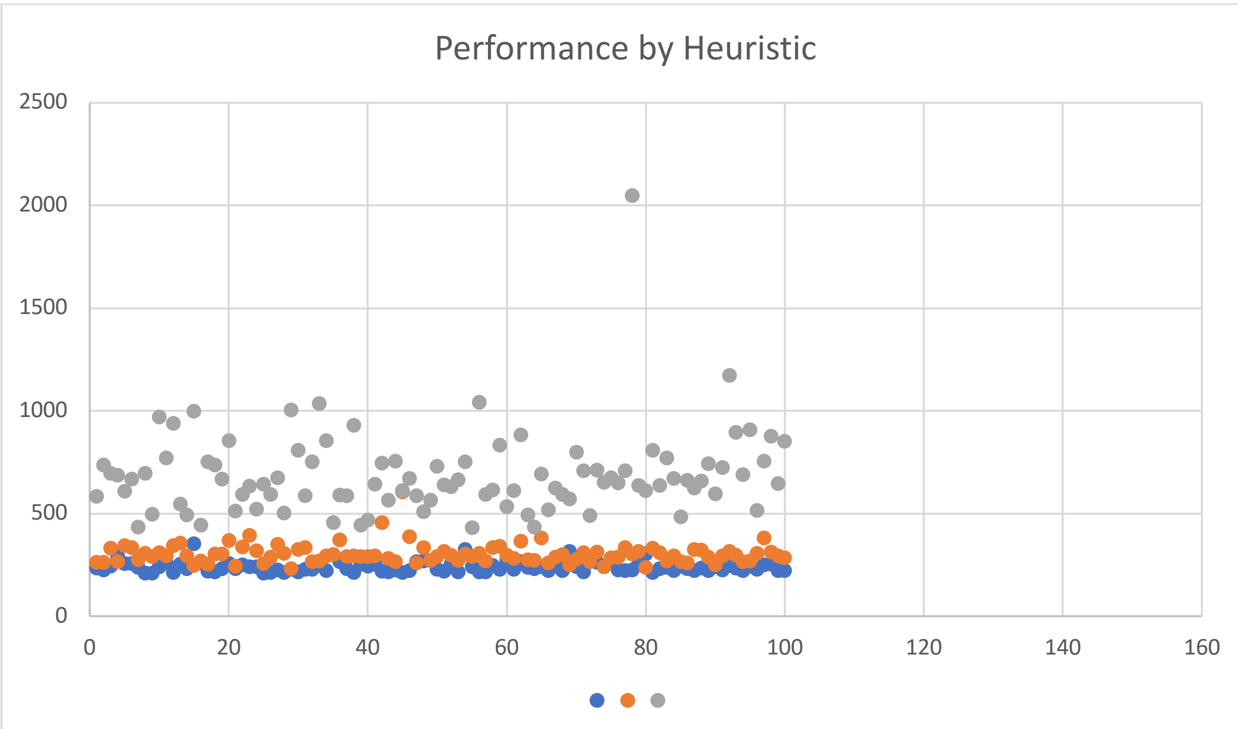
2a. For each increment of 0.05 in probability, I generated 100 gridworlds of dimension 100 and checked for solvability. As we can see in the data below, there is a sharp decline between 0.35 and .4. Also, the halfway point seems to be slightly before 0.3, as more gridworlds are unsolvable past this point than solvable.

Probabilities	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
0.05	100	100	100	98	100
0.1	96	98	99	96	97
0.15	93	94	88	88	89
0.2	83	87	80	82	89
0.25	67	68	68	74	72
0.3	49	49	42	45	49
0.35	24	29	24	21	28
0.4	1	1	0	1	3
0.45	0	0	0	0	0
0.5	0	0	0	0	0
0.55	0	0	0	0	0
0.6	0	0	0	0	0
0.65	0	0	0	0	0
0.7	0	0	0	0	0
0.75	0	0	0	0	0
0.8	0	0	0	0	0
0.85	0	0	0	0	0
0.9	0	0	0	0	0
0.95	0	0	0	0	0



2b. Rather than running A*, as we are only looking for a solution and don't care if it's optimal, depth first search that goes down or right before up or left would progress more rapidly towards the goal. Alternatively we could run a seach on blocked tiles that includes diagonal movement, as a blocked path would necessitate a path of blocked tiles that go across the board.

3a. I performed 100 solvable tests at 0.25 probability for each heuristic type and counted the amount of nodes expanded by each. Euclidean performed the best, with Manhattan distance not too far from it, and Chebyshev way behind. In 100 trials, a Euclidean metric resulted in expanding an average of 241.03 nodes, a Manhattan metric resulted in expanding an average of 303.73 nodes, while a Chebyshev metric resulted in expanding an average of 689.41 nodes.



3b. He is correct. Let's take h=max(h1,h2) as our heuristic. At a given state s, let's say WLOG that h1 is bigger. Then as h1 is consistent, if any neighbors s' take on the value of h1 then the inequality h(s) ≤ c(s, a, s') + h(s') must be true. Let's now consider if a neighbor s'' takes on the value of h2. Since we know that h1 is consistent, we know that h1(s)≤c(s,a,s'')+h1(s''). We also know that h1(s'')<h2(s''), as h=max(h1,h2), so by transitive property, h1(s)-c(s,a,s'')≤h2(s''), or h1(s)≤c(s,a,s'')+h2(s'').

Now let's take h=min(h1,h2) as our heuristic. At a given state s, let's say WLOG that h1 is smaller. Then as h1 is consistent, if any neighbors s' take on the value of h1 then the inequality h(s) ≤ c(s, a, s') + h(s') must be true. Let's now consider if a neighbor s'' takes on the value of h2. Since we know that h2 is consistent, we know that h2(s)≤c(s,a,s'')+h2(s''). We also know that h1(s)<h2(s), as h=min(h1,h2), so by transitive property, h1(s)≤ c(s,a,s'')+h2(s'').