Comp 424 Assignment 1

Jonathan Pearce, 260672004

January 30, 2018

Problem 1a. A solution path is represented by the sequence of pieces that were moved in order to reach the goal state
i) Breadth First Search Solution Path: $3 \rightarrow 4 \rightarrow 1$
ii) Uniform Cost Search Solution Path: $3 \rightarrow 4 \rightarrow 1$
iii) Depth First Search Solution Path: $2 \rightarrow 4 \rightarrow 1 \rightarrow 5 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 5 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 4 \rightarrow 1 \rightarrow 2 \rightarrow 5 \rightarrow 3 \rightarrow 2 \rightarrow 5 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 1 \rightarrow 5 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 1 \rightarrow 5 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 5 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 4 \rightarrow 1 \rightarrow 2 \rightarrow 5 \rightarrow 3 \rightarrow 2 \rightarrow 5 \rightarrow 3 \rightarrow 2 \rightarrow 5 \rightarrow 1 \rightarrow 4 \rightarrow 5 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 1 \rightarrow 5 \rightarrow 3 \rightarrow 1 \rightarrow 5 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 5 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 1 \rightarrow 5 \rightarrow 3 \rightarrow 1 \rightarrow 5 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 5 \rightarrow 3 \rightarrow 2 \rightarrow 5 \rightarrow 3 \rightarrow 2 \rightarrow 5 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 3 \rightarrow 1 \rightarrow 2 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1$
iv) Iterative Deepening Solution Path: $3 \rightarrow 4 \rightarrow 1$

Problem 1b.

$$h_2(n) = \sum_{i=1}^{5} \text{manhattan distance(i, n)}$$

$$= \text{minimum number of moves required to reach goal state}$$

Since all moves cost >= 1, it follows

$$h_2(n) <= \text{cost of minimum number of moves required to reach goal state}$$

$$= \text{minimum cost to reach goal state}$$

$$= h^*(n)$$

Therefore $h_2(n)$, the Manhattan distance heuristic is admissible for A* search.

Problem 1c. We define a new heuristic, h_3 as follows,

$$h_3(n) = \sum_{i=1}^{5} i * \text{manhattan distance(i, n)}$$

Since moving piece i costs i, it follows,

$$h_3(n) = \text{cost of minimum number of moves required to reach goal state}$$
$$= \text{minimum cost to reach goal state}$$
$$= h^*(n)$$

Therefore $h_3(n)$ is admissible. Since $i >= 1$ we also have,

$$\sum_{i=1}^{5} i * \text{manhattan distance(i, n)} >= \sum_{i=1}^{5} \text{manhattan distance(i, n)}$$
$$h_3(n) >= h_2(n)$$

Therefore $h_3(n)$ dominates $h_2(n)$.

Problem 1d. Manhattan Distance is not admissible in this scenario

Initial State n:

5	1	2
0	4	3

$$h_2(n) = \sum_{i=1}^{5} \text{manhattan distance(i, n)} = 1$$

Swap 0 and 5 (cost 0.5)

0	1	2
5	4	3

The above grid is the goal state and it cost 0.5 to reach this state from the initial configuration. However, at the initial configuration the heuristic had value 1 and therefore overestimated the cost of the solution path. Therefore it is not an admissible heuristic.

Problem 2a. Consider a state space where every node has only one child and there exists one solution path. The goal state is at depth n in the search tree.

Runtime DFS: Depth-First Search will simply proceed directly through the search tree (only one path) until reaching the solution $O(n)$.

Runtime IDS: Iterative Deepening will proceed directly through the search tree to a depth of 1. Then it will repeat for a depth of 2, it will continue until reaching the solution at depth n

$$= 1 + 2 + ...n = \frac{n(n+1)}{2}$$

Therefore runtime is $O(n^2)$

Therefore this is a state space where depth first search performs much better than iterative deepening search.

Problem 2b. Suppose all operators have cost $= k$, where $k > 0$. Then, any path in the search tree to depth n has cost nk, and any path in the search tree to depth $n + 1$ has cost $(n + 1)k$. Since uniform-cost search uses a priority queue every path to depth n will be explored before any path to depth $n+1$, since the cost of the path is less $(nk < (n+1)k)$. This is equivalent to Breadth-First Search and thus BFS is special case of uniform-cost search.

Problem 2c. Suppose the heuristic of a node at depth n is $-n$ (i.e all nodes at depth 5 have a heuristic value of -5). Because best-first search favours lower heuristic values, in this scenario best-first search will always prefer moving deeper into the search tree rather than exploring more states at its current depth. This is the exact behaviour of depth-first search and therefore DFS is a special case of best-first search.

Problem 2d. Suppose $h(n) = 0 \ \forall n$. It follows that the evaluation function for A* reduces to $f(n) = g(n)$ where $g(n)$ is the cost to reach the node. Now A* performs a greedy search only based on the cost up to the node, this is exactly the concept behind uniform-cost search, where we only consider the cost-so-far. We conclude that uniform cost search is a special case of A*.

Problem 3. Question 3 is at the end of the pdf

Problem 4a. Let R_i^C represent the column of the ith rook and let R_i^R represent the row of the ith rook, therefore the pair (R_i^C, R_i^R) defines the location on the board of the ith rook. The constraint satisfaction problem can be defined as follows.

Variables: $\{R_1^C, R_1^R, R_2^C, R_2^R..., R_k^C, R_k^R\}$
Domain (same for all variables): $\{1, 2, 3, ..., n\}$
Constraints:

- $R_i^C \neq R_j^C$, $\forall i, j$ where $i \neq j$ (cannot be in the same column)
- $R_i^R \neq R_j^R$, $\forall i, j$ where $i \neq j$ (cannot be in the same row)

Note: If k > n, then there are more rooks than there are columns on the board and it is clear from the pigeon principle that no solution is possible. However for the sake of the exercise I did not consider this condition and instead worked through the complete constraint satisfaction problem for practice.

Problem 4b. Graph below

Problem 4c. Graph below

4) b)

	R^c	R^R
R_1	1,2,3	1,2,3
R_2	1,2,3	1,2,3
R_3	1,2,3	1,2,3

$R_1^c = 1$

	R^c	R^R
R_1	1	1,2,3
R_2	1,2,3	1,2,3
R_3	1,2,3	1,2,3

$R_1^R = 1$

	R^c	R^R
R_1	1	1
R_2	1,2,3	1,2,3
R_3	1,2,3	1,2,3

$R_2^c = 1$

	R^c	R^R
R_1	1	1
R_2	1	1,2,3
R_3	1,2,3	1,2,3

STOP! $R_1^c = R_2^c$

$R_2^c = 2$

	R^c	R^R
R_1	1	1
R_2	2	1,2,3
R_3	1,2,3	1,2,3

$R_2^R = 1$

	R^c	R^R
R_1	1	1
R_2	2	1
R_3	1,2,3	1,2,3

STOP! $R_1^R = R_2^R$

$R_2^R = 2$

	R^c	R^R
R_1	1	1
R_2	2	2
R_3	1,2,3	1,2,3

$R_3^c = 1$

	R^c	R^R
R_1	1	1
R_2	2	2
R_3	1	1,2,3

STOP $R_1^c = R_3^c$

$R_3^c = 2$

	R^c	R^R
R_1	1	1
R_2	2	2
R_3	2	1,2,3

STOP $R_1^c = R_2^c$

$R^3_c = 3$

	R^c	R^R
R_1	1	1
R_2	2	2
R_3	3	1,2,3

$R_3^R = 1$

	R^c	R^R
R_1	1	1
R_2	2	2
R_3	3	1

STOP! $R_1^R = R_3^R$

$R_3^R = 2$

	R^c	R^R
R_1	1	1
R_2	2	2
R_3	3	2

STOP! $R_2^R = R_3^R$

$R_3^R = 3$

	R^c	R^R
R_1	1	1
R_2	2	2
R_3	3	3

Solution!!!

4) c)

	R^c	R^R
R_1	1,2,3	1,2,3
R_2	1,2,3	1,2,3
R_3	1,2,3	1,2,3

$R_1^c = 1$

	R^c	R^R
R_1	1	1,2,3
R_2	2,3	1,2,3
R_3	2,3	1,2,3

$R_1^R = 1$

	R^c	R^R
R_1	1	1
R_2	2,3	2,3
R_3	2,3	2,3

$R_2^c = 2$

	R^c	R^R
R_1	1	1
R_2	2	2,3
R_3	3	2,3

$R_2^R = 2$

	R^c	R^R
R_1	1	1
R_2	2	2
R_3	3	3

$R_3^c = 3$

	R^c	R^R
R_1	1	1
R_2	2	2
R_3	3	3

$R_3^R = 3$

	R^c	R^R
R_1	1	1
R_2	2	2
R_3	3	3

Solution!!

Q3a: Hill Climbing

Final X Coordinates

		Step Size									
		0.01	**0.02**	**0.03**	**0.04**	**0.05**	**0.06**	**0.07**	**0.08**	**0.09**	**0.1**
Starting Points	**0**	1.74	1.74	1.74	1.72	1.75	1.74	1.75	1.76	1.71	1.70
	1	1.74	1.74	1.75	1.72	1.75	1.72	1.77	1.72	1.72	1.70
	2	1.74	1.74	1.73	1.72	1.75	1.76	1.72	1.76	1.73	1.70
	3	1.74	1.74	1.74	1.72	1.75	1.74	1.74	1.72	1.74	1.70
	4	3.96	3.96	3.97	3.96	3.95	3.94	3.93	3.92	4.00	4.00
	5	5.32	5.32	5.33	5.32	5.30	5.30	5.35	5.32	5.36	5.30
	6	6.39	6.38	6.39	6.40	6.40	6.36	6.42	6.40	6.36	6.40
	7	7.31	7.30	7.30	7.32	7.30	7.30	7.28	7.32	7.27	7.30
	8	8.12	8.12	8.12	8.12	8.10	8.12	8.14	8.16	8.09	8.10
	9	8.86	8.86	8.85	8.88	8.85	8.88	8.86	8.84	8.82	8.90
	10	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00

Final Y Coordinates

		Step Size									
		0.01	**0.02**	**0.03**	**0.04**	**0.05**	**0.06**	**0.07**	**0.08**	**0.09**	**0.1**
Starting Points	**0**	0.396	0.396	0.396	0.396	0.396	0.396	0.396	0.396	0.396	0.395
	1	0.396	0.396	0.396	0.396	0.396	0.396	0.395	0.396	0.396	0.395
	2	0.396	0.396	0.396	0.396	0.396	0.396	0.396	0.396	0.396	0.395
	3	0.396	0.396	0.396	0.396	0.396	0.396	0.396	0.396	0.396	0.395
	4	0.334	0.334	0.334	0.334	0.334	0.333	0.332	0.330	0.330	0.330
	5	0.310	0.310	0.310	0.310	0.310	0.310	0.305	0.310	0.302	0.310
	6	0.296	0.296	0.296	0.295	0.295	0.291	0.291	0.295	0.291	0.295
	7	0.286	0.285	0.285	0.285	0.285	0.285	0.280	0.285	0.275	0.285
	8	0.278	0.278	0.278	0.278	0.273	0.278	0.275	0.265	0.269	0.273
	9	0.271	0.271	0.270	0.268	0.270	0.268	0.271	0.266	0.253	0.256
	10	-0.069	-0.069	-0.069	-0.069	-0.069	-0.069	-0.069	-0.069	-0.069	-0.069

Iterations

		Step Size									
		0.01	**0.02**	**0.03**	**0.04**	**0.05**	**0.06**	**0.07**	**0.08**	**0.09**	**0.1**
Starting Points	**0**	175	88	59	44	36	30	26	23	20	18
	1	75	38	26	19	16	13	12	10	9	8
	2	27	14	10	8	6	5	5	4	4	4
	3	127	64	43	33	26	22	19	17	15	14
	4	5	3	2	2	2	2	2	2	1	1
	5	33	17	12	9	7	6	6	5	5	4
	6	40	20	14	11	9	7	7	6	5	5
	7	32	16	11	9	7	6	5	5	4	4
	8	13	7	5	4	3	3	3	3	2	2
	9	15	8	6	4	4	3	3	3	3	2
	10	1	1	1	1	1	1	1	1	1	1

Q3a: Simulated Annealing (Discussion after charts)

Step Size = 0.05

Final X Coordinates

	Cooling Rate	0.5			0.75			0.9			0.99		
	Temperature	**5**	**500**	**50000**	**5**	**500**	**50000**	**5**	**500**	**50000**	**5**	**500**	**50000**
Starting Points	**0**	0.15	0.15	0.40	0.35	0.40	0.75	0.70	0.65	0.90	1.75	1.75	1.75
	1	1.05	1.10	1.20	1.30	1.30	1.00	1.15	1.40	1.45	1.75	1.75	1.75
	2	2.00	1.75	1.75	1.85	1.75	1.75	1.90	1.95	1.75	1.75	1.75	1.90
	3	3.00	2.50	2.80	3.55	2.70	2.70	2.70	1.75	2.35	1.75	3.95	1.75
	4	4.00	3.95	3.95	3.95	3.95	3.95	3.95	3.95	3.95	3.95	3.95	4.00
	5	5.00	5.10	5.15	5.30	5.30	5.15	5.30	5.30	5.30	5.30	3.95	3.95
	6	6.05	6.00	6.25	6.30	5.55	6.40	5.30	5.30	6.40	5.30	3.95	6.40
	7	7.10	7.10	7.10	7.05	7.30	6.40	7.30	6.40	6.40	6.40	7.30	6.40
	8	8.10	8.00	8.00	8.10	8.10	8.10	8.10	7.30	8.10	8.10	8.10	6.40
	9	8.85	9.50	8.85	9.55	9.55	8.85	8.85	8.85	8.85	8.85	9.55	6.40
	10	10.00	10.00	10.00	9.55	10.00	10.00	10.00	9.55	9.55	9.55	6.40	8.10

Final Y Coordinates

	Cooling Rate	0.5			0.75			0.9			0.99		
	Temperature	**5**	**500**	**50000**	**5**	**500**	**50000**	**5**	**500**	**50000**	**5**	**500**	**50000**
Starting Points	**0**	0.005	0.005	0.037	0.029	0.037	0.123	0.109	0.095	0.172	0.396	0.396	0.396
	1	0.224	0.242	0.277	0.311	0.311	0.206	0.260	0.341	0.355	0.396	0.396	0.396
	2	0.352	0.396	0.396	0.389	0.396	0.396	0.380	0.368	0.396	0.396	0.396	0.380
	3	-0.348	0.006	-0.254	0.006	-0.176	-0.176	-0.176	0.396	0.139	0.396	0.334	0.396
	4	0.330	0.334	0.334	0.334	0.334	0.334	0.334	0.334	0.334	0.334	0.334	0.330
	5	-0.021	0.133	0.200	0.310	0.310	0.200	0.310	0.310	0.310	0.310	0.334	0.334
	6	-0.157	-0.226	0.188	0.249	0.093	0.295	0.310	0.310	0.295	0.310	0.334	0.295
	7	0.021	0.021	0.021	-0.080	0.285	0.295	0.285	0.295	0.295	0.295	0.285	0.295
	8	0.273	0.154	0.154	0.273	0.273	0.273	0.273	0.285	0.273	0.273	0.273	0.295
	9	0.270	0.242	0.270	0.266	0.266	0.270	0.270	0.270	0.270	0.270	0.266	0.295
	10	-0.069	-0.069	-0.069	0.266	-0.069	-0.069	-0.069	0.266	0.266	0.266	0.295	0.273

Iterations

	Cooling Rate	0.5			0.75			0.9			0.99		
	Temperature	**5**	**500**	**50000**	**5**	**500**	**50000**	**5**	**500**	**50000**	**5**	**500**	**50000**
Starting Points	**0**	9	16	23	22	38	54	59	103	147	619	1077	1535
	1	9	16	23	22	38	54	59	103	147	619	1077	1535
	2	9	16	23	22	38	54	59	103	147	619	1077	1535
	3	9	16	23	22	38	54	59	103	147	619	1077	1535
	4	9	16	23	22	38	54	59	103	147	619	1077	1535
	5	9	16	23	22	38	54	59	103	147	619	1077	1535
	6	9	16	23	22	38	54	59	103	147	619	1077	1535
	7	9	16	23	22	38	54	59	103	147	619	1077	1535
	8	9	16	23	22	38	54	59	103	147	619	1077	1535
	9	9	16	23	22	38	54	59	103	147	574	1048	1502
	10	9	16	23	22	38	54	45	103	147	595	1077	1504

Step Size = 0.10

Final X Coordinates

	Cooling Rate	0.5			0.75			0.9			0.99		
	Temperature	**5**	**500**	**50000**	**5**	**500**	**50000**	**5**	**500**	**50000**	**5**	**500**	**50000**
Starting Points	**0**	0.30	0.20	0.50	0.60	0.40	1.20	1.20	1.50	1.70	1.70	1.70	1.70
	1	1.10	1.50	1.30	1.20	1.70	1.60	1.70	1.20	1.70	1.70	1.70	1.70
	2	1.80	1.70	1.90	1.70	1.70	1.70	1.70	1.70	1.70	1.70	1.70	1.70
	3	3.20	3.60	3.30	3.50	4.00	1.70	2.10	1.80	1.70	1.70	1.70	1.70
	4	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	1.70	4.00	1.70
	5	5.00	5.00	5.30	5.30	5.30	5.30	4.00	5.30	4.00	4.00	4.00	1.70
	6	6.20	6.40	6.20	6.40	6.40	5.30	6.40	5.30	5.30	6.40	4.00	4.00
	7	6.50	7.20	7.30	7.30	7.30	6.40	6.40	7.30	7.30	7.30	4.00	5.30
	8	8.00	8.10	8.10	8.10	7.30	7.30	8.10	7.30	8.10	7.30	5.30	5.30
	9	8.90	8.90	8.90	8.90	8.90	8.90	8.90	9.50	7.30	8.10	8.90	8.10
	10	9.50	9.50	9.50	9.50	8.90	9.50	9.50	9.50	8.90	7.30	7.30	7.30

Final Y Coordinates

	Cooling Rate	0.5			0.75			0.9			0.99		
	Temperature	**5**	**500**	**50000**	**5**	**500**	**50000**	**5**	**500**	**50000**	**5**	**500**	**50000**
Starting Points	**0**	0.021	0.010	0.057	0.081	0.037	0.277	0.277	0.367	0.395	0.395	0.395	0.395
	1	0.242	0.367	0.311	0.277	0.395	0.385	0.395	0.277	0.395	0.395	0.395	0.395
	2	0.394	0.395	0.380	0.395	0.395	0.395	0.395	0.395	0.395	0.395	0.395	0.395
	3	-0.322	0.067	-0.259	-0.054	0.330	0.395	0.309	0.394	0.395	0.395	0.395	0.395
	4	0.330	0.330	0.330	0.330	0.330	0.330	0.330	0.330	0.330	0.395	0.330	0.395
	5	-0.021	-0.021	0.310	0.310	0.310	0.310	0.330	0.310	0.330	0.330	0.330	0.395
	6	0.108	0.295	0.108	0.295	0.295	0.310	0.295	0.310	0.310	0.295	0.330	0.330
	7	0.225	0.203	0.285	0.285	0.285	0.295	0.295	0.285	0.285	0.285	0.330	0.310
	8	0.154	0.273	0.273	0.273	0.285	0.285	0.273	0.285	0.273	0.285	0.310	0.310
	9	0.256	0.256	0.256	0.256	0.256	0.256	0.256	0.242	0.285	0.273	0.256	0.273
	10	0.242	0.242	0.242	0.242	0.256	0.242	0.242	0.242	0.256	0.285	0.285	0.285

Iterations

	Cooling Rate	0.5			0.75			0.9			0.99		
	Temperature	**5**	**500**	**50000**	**5**	**500**	**50000**	**5**	**500**	**50000**	**5**	**500**	**50000**
Starting Points	**0**	9	16	23	22	38	54	59	103	147	619	1077	1533
	1	9	16	23	22	38	54	59	103	147	619	1077	1535
	2	9	16	23	22	38	54	59	103	147	619	986	1535
	3	9	16	23	22	38	54	59	103	147	619	1077	1535
	4	9	16	23	22	38	54	59	103	147	619	1077	1535
	5	9	16	23	22	38	54	59	103	147	619	1077	1535
	6	9	16	23	22	38	54	59	103	147	474	974	1408
	7	9	16	23	22	38	54	59	100	145	553	974	1397
	8	9	16	23	22	38	54	59	103	147	483	962	1459
	9	9	16	23	22	38	54	59	103	146	428	1055	1447
	10	9	16	23	22	38	54	59	103	147	446	880	1473

Simulated Annealing Discussion:

For part b of question 3 I chose to use 2 different step sizes. Simulated Annealing is used to escape local maxima and minima in order to try and find the global maximum by occasionally taking non-optimal moves in an attempt to search new areas of the state space. The issue with small step sizes such 0.01 is that even when you take a non-optimal move your not going to move very far in the state space (in our example the x axis) and thus it is very difficult to get over and escape from local maxima and minima. I choose 0.1 because it was the largest step size we could select which makes moving through the state space easiest, and I choose 0.05 to compare with 0.1, in order to confirm my idea that a larger step size is more successful getting past local maxima in search of the global maxima.

From the results it is clear that the step size of 0.1 produced better results than the step size of 0.05. The 0.1 step size was only kept below the y-axis (i.e. the best value was negative) 5 times, where as this happened 15 times for the 0.05 step size. As well when starting at x=10, the 0.05 step size was only able to escape from the local minimum just before x=10, 6 times where as step size 0.1 was able to move past this initial local minimum all 12 times (all combinations of cooling rate and temperature). In this situation the larger step size was more successful.

Cooling rates and temperatures were selected to give a good range of values for both categories. It was interesting to see how critical it is to start with a high temperature to allow enough 'bad' moves at the beginning of an experiment. The difference between temperature 5 and temperature 50,000 is a factor of 10,000 however with the exponential growth of the cooling rate this factor of 10,000 only produced approximately 3 times as many iterations per experiment. Also its clear from the iteration charts that the cooling rate should most likely be much closer to 1 than to 0 if you want a reasonable amount of iterations, for this simple case ~50 or ~100 iterations worked fine, but for more complex scenarios we would need a cooling rate of 0.9 or 0.99.

One other observation from the final Y charts was that more iterations does not always vastly improve results, the experiments with cool = 0.9 and temp = 500 (~100 iterations) performed just about as well as experiments with cool = 0.99 and temp = 50,000 (~1500 iterations).

Note on Iterations: in my code I included a second condition such that after 10 iterations if there was no new position found, the annealing would stop. This had a more significant and noticeable effect with the 0.1 step size.