Q1-1: Consider running a genetic algorithm with a mutation probability of 0. Is the mean population fitness at iteration *i*+1 expected to be greater than mean population fitness at iteration *i*?

1. No

2. Yes

3. Unsure

Q1-1: Consider running a genetic algorithm with a mutation probability of 0. Is the mean population fitness at iteration *i*+1 expected to be greater than mean population fitness at iteration *i*?

1. No
2. Yes
3. Unsure

Q1-2: Consider running a genetic algorithm with a mutation probability of 0. Is the mean population fitness at iteration *i*+1 always guaranteed to be greater than mean population fitness at iteration *i*?

1. No

2. Yes

3. Unsure

Q1-2: Consider running a genetic algorithm with a mutation probability of 0. Is the mean population fitness at iteration *i*+1 always guaranteed to be greater than mean population fitness at iteration *i*?

1. No

2. Yes

3. Unsure

# Q1-3: What could go wrong in a genetic algorithm if the mutation probability is 0.95?

1. The population would converge to similar states too quickly
2. The algorithm would randomly explore the state space
3. The cross-over operation would be too slow
4. We would be unable to calculate fitness scores

# Q1-3: What could go wrong in a genetic algorithm if the mutation probability is 0.95?

1. The population would converge to similar states too quickly
2. The algorithm would randomly explore the state space
3. The cross-over operation would be too slow
4. We would be unable to calculate fitness scores

# Q1-4: What could go wrong in a genetic algorithm if the initial population contains the same state *N* times?

1. The population would converge to similar states too quickly
2. The algorithm would randomly explore the state space
3. The cross-over operation would be too slow
4. We would be unable to calculate fitness scores

# Q1-4: What could go wrong in a genetic algorithm if the initial population contains the same state *N* times?

1. The population would converge to similar states too quickly
2. The algorithm would randomly explore the state space
3. The cross-over operation would be too slow
4. We would be unable to calculate fitness scores

Q2-1: Consider the Rubik's cube puzzle. The goal is to rotate the cube so that all 6 faces have tiles of the same color. Which is an admissible heuristic?

1. The worst case number of moves needed to reach the goal from any initial state
2. 0
3. The number of moves needed to solve the face with a blue center tile
4. 1 and 2
5. 2 and 3

Image from https://www.fiverr.com/darkdragon532/teach-you-how-to-solve-a-rubiks-cube-3x3

Q2-1: Consider the Rubik's cube puzzle. The goal is to rotate the cube so that all 6 faces have tiles of the same color. Which is an admissible heuristic?

1. The worst case number of moves needed to reach the goal from any initial state
2. 0
3. The number of moves needed to solve the face with a blue center tile
4. 1 and 2
5. 2 and 3

Image from https://www.fiverr.com/darkdragon532/teach-you-how-to-solve-a-rubiks-cube-3x3

Q2-2: Consider the Rubik's cube puzzle. The goal is to rotate the cube so that all 6 faces have tiles of the same color. Which is an admissible heuristic?

1. The number of moves required if we can only rotate the faces up or to the right.
2. The number of misplaced edge tiles
3. The number of misplaced edge tiles on the blue face minus the number of misplaced edge tiles on the orange face

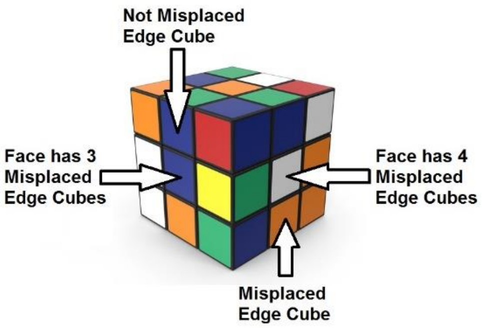


Image from https://www.cs.huji.ac.il/~ai/projects/2017/heuristic_%20search/learning_heuristics_for_the_rubiks_cube/

Q2-2: Consider the Rubik's cube puzzle. The goal is to rotate the cube so that all 6 faces have tiles of the same color. Which is an admissible heuristic?

1. The number of moves required if we can only rotate the faces up or to the right.
2. The number of misplaced edge tiles
3. The number of misplaced edge tiles on the blue face minus the number of misplaced edge tiles on the orange face

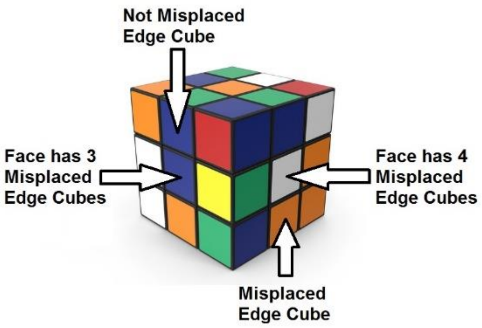


Image from https://www.cs.huji.ac.il/~ai/projects/2017/heuristic_%20search/learning_heuristics_for_the_rubiks_cube/

Q2-3: $h_1$ and $h_2$ are admissible heuristics. Which of the following are also admissible?

1. $\max(h_1, h_2)$

2. $h_1 + h_2$

3. $(1-c)*h_1 + c*h_2$, $c$ in [0, 1]

4. 1 and 3

5. All of the above

Q2-3: $h_1$ and $h_2$ are admissible heuristics. Which of the following are also admissible?

1. max($h_{1,}$ $h_2$)
2. $h_1 + h_2$
3. (1-$c$)*$h_1$ + $c$*$h_2$, $c$ in [0, 1]
4. 1 and 3
5. All of the above

# Q3-1: Which of the following is likely to give the best Temp schedule for simulated annealing?

1. $Temp_{t+1} = Temp_t * 1.25$

2. $Temp_{t+1} = Temp_t$

3. $Temp_{t+1} = Temp_t * 0.8$

4. $Temp_{t+1} = Temp_t * 0.0001$

# Q3-1: Which of the following is likely to give the best Temp schedule for simulated annealing?

1. $Temp_{t+1} = Temp_t * 1.25$

2. $Temp_{t+1} = Temp_t$

3. $Temp_{t+1} = Temp_t * 0.8$

4. $Temp_{t+1} = Temp_t * 0.0001$

# Q3-2: Which of the following would be better to solve with simulated annealing than A* search?

1. Finding the smallest set of vertices in a graph that involve all edges
2. Finding the fastest way to schedule jobs with varying runtimes on machines with varying processing power
3. Finding the fastest way through a maze
4. 1 and 2
5. 2 and 3

Q3-2: Which of the following would be better to solve with simulated annealing than A* search?

1. Finding the smallest set of vertices in a graph that involve all edges
2. Finding the fastest way to schedule jobs with varying runtimes on machines with varying processing power
3. Finding the fastest way through a maze
4. 1 and 2
5. 2 and 3