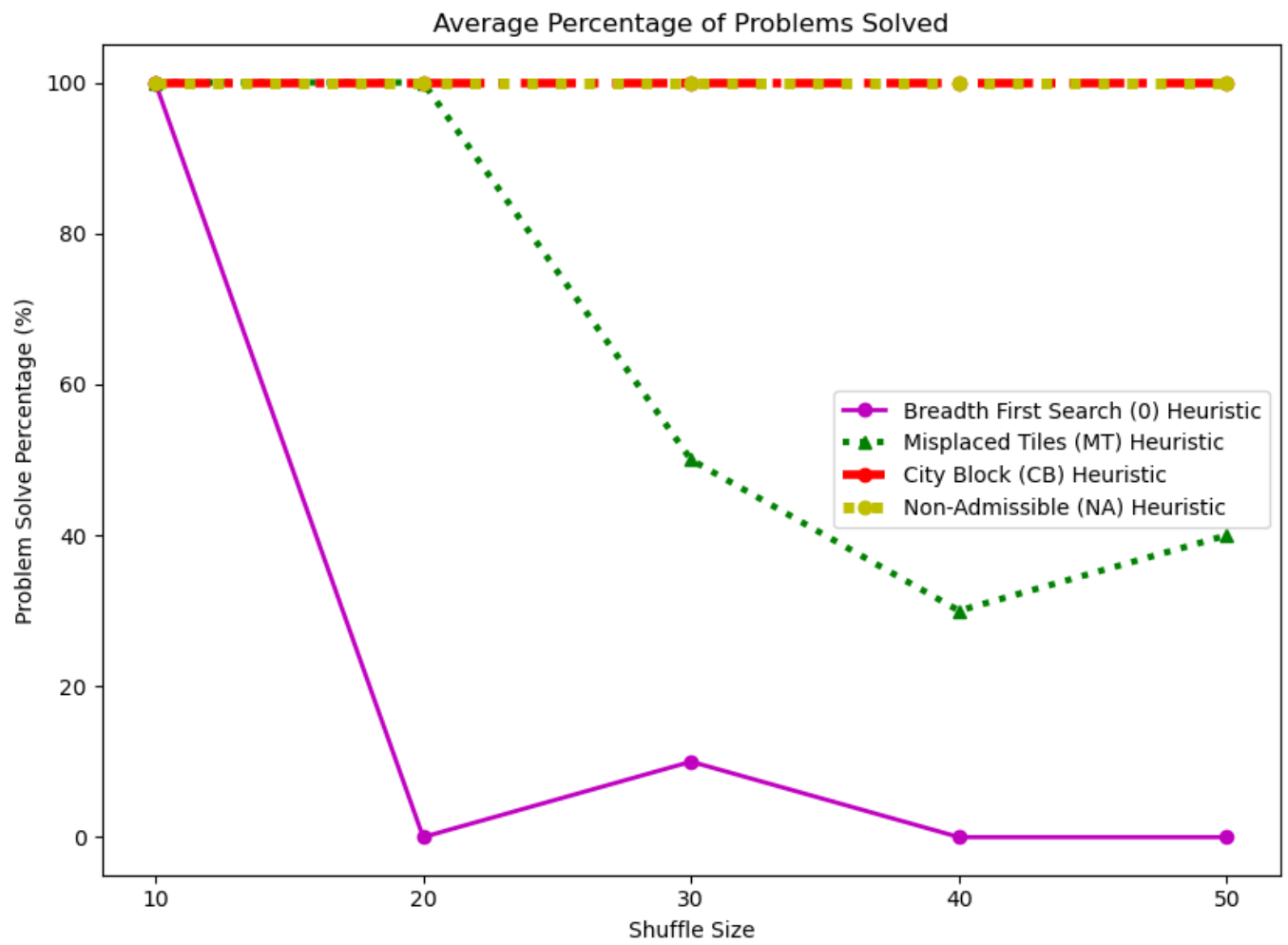
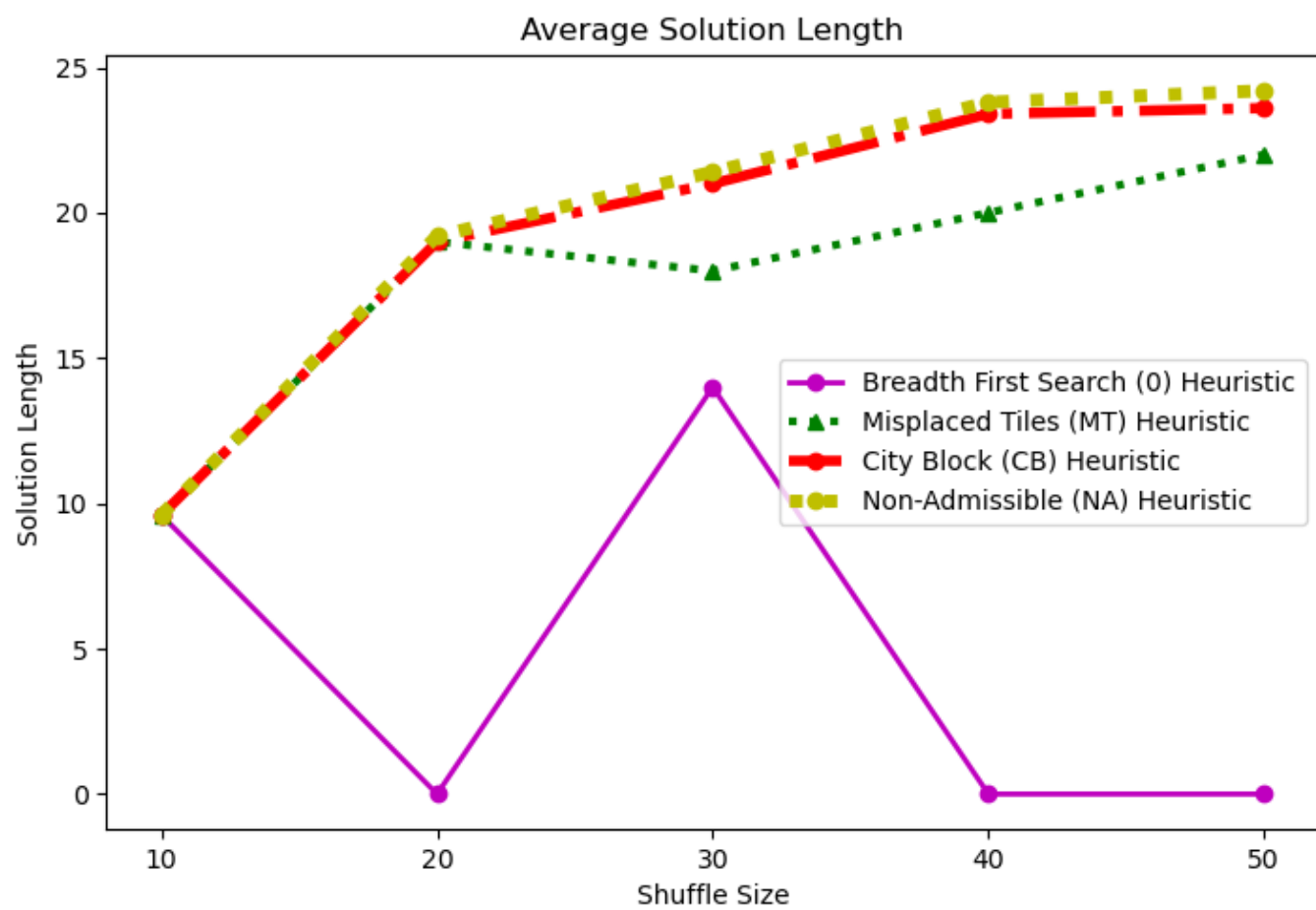
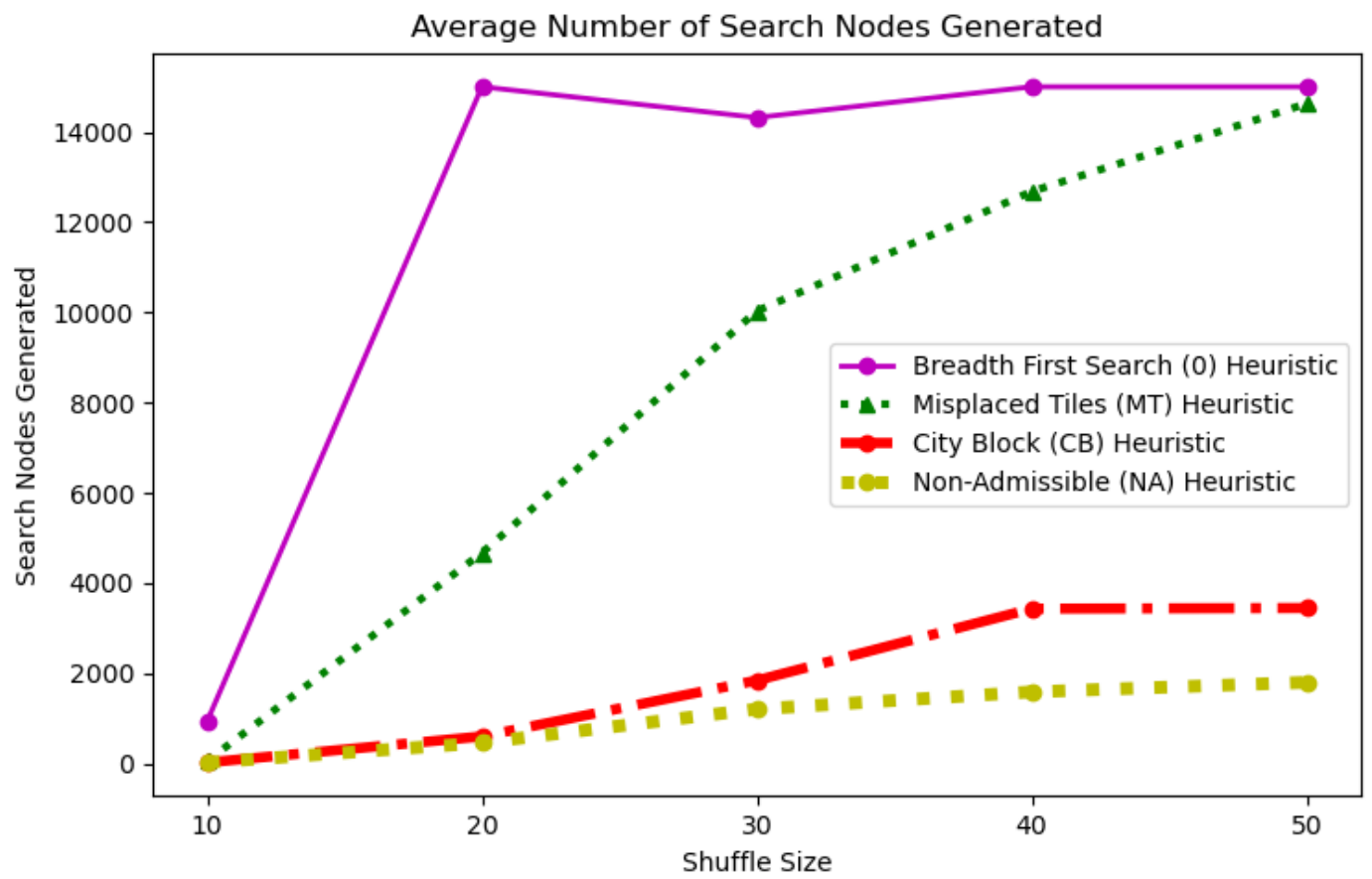
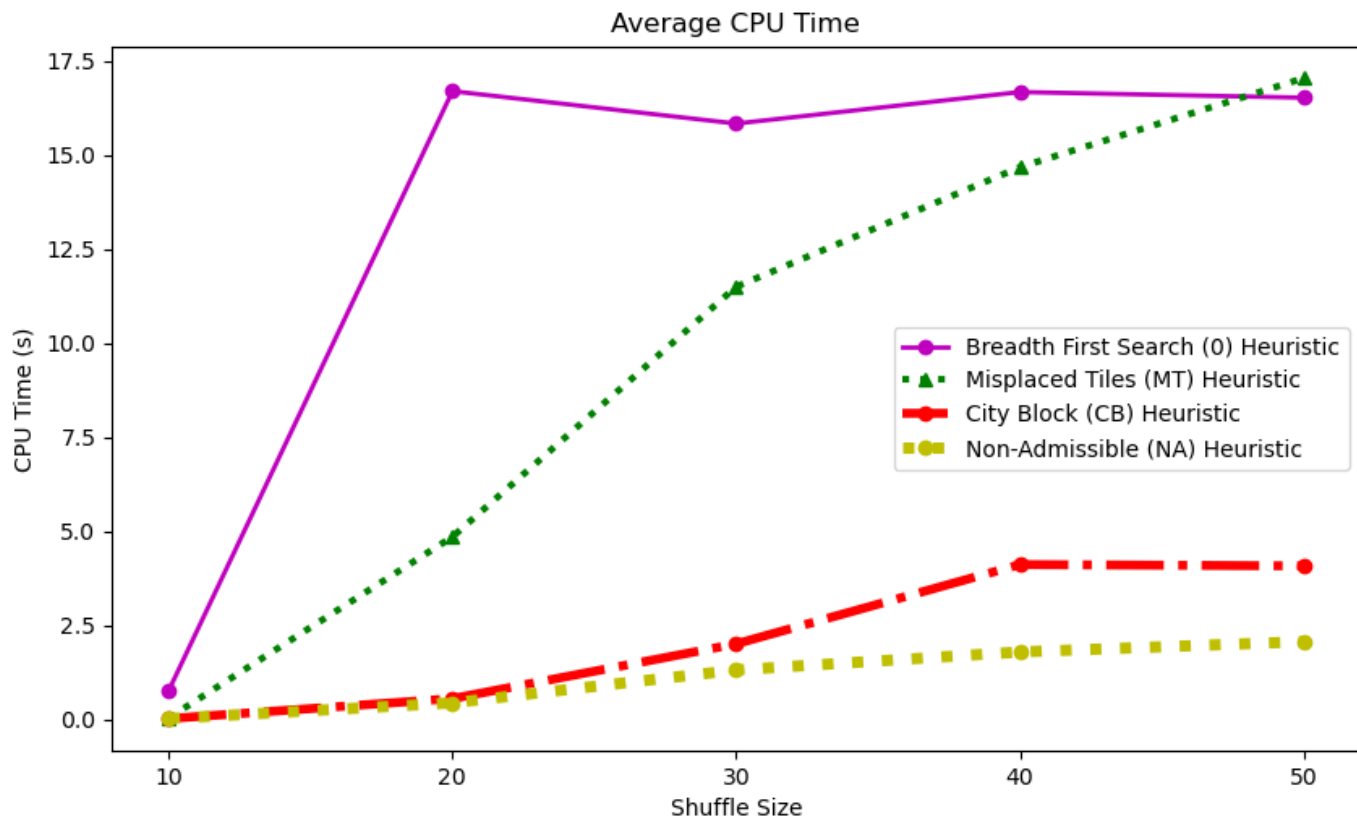


Plots









8 Puzzle Report

I used a 15,000-search node limit for my plots and Matplotlib to create the plots.

1. How did the two algorithms and the three heuristics compare in terms of the number of nodes searched, the solution length, and the CPU time?

For number of nodes searched, BFS generated the most nodes on average by far, generally reaching the 15,000 node limit for shuffle sizes ≥ 20 . MT was second highest, followed by CB for shuffles ≥ 20 and NA had the least amount of search nodes generated.

For solution lengths, when solved percentage rates were the same (100%) between admissible heuristics, average solution lengths between the admissible heuristics were the same. The NA heuristic could get optimal solutions or suboptimal (longer-than-optimal) solutions, especially for larger shuffle sizes.

For CPU time, heuristics that explored the most nodes, like BFS, took the longest, and then more accurate heuristics took less and less time because of fewer nodes searched. BFS took the longest, followed by MT, CB, and then NA was the fastest.

2. Is there a clear preference ordering among the heuristics? Is there a tradeoff between solution length and the number of nodes searched or between CPU time and the number of nodes searched?

There's a clear ordering preference among the heuristics, where the NA heuristic has the best space/time complexities for large shuffle sizes, followed by CB, MT, and then BFS. For tradeoffs, we can reduce the number of nodes searched through a NA heuristic, but we end up with a longer solution length on average (suboptimal solution). Relatedly, when more time is spent finding a solution, more memory is being used because of the increasing number of search nodes generated.

3. How did you come up with your heuristic evaluation function?

To come up with my heuristic evaluation function, I thought about which heuristics could overestimate the actual solution to the goal state. While CB and MT are admissible heuristics, I noticed together they create a heuristic that could overestimate the actual length to the goal in different situations. So, I added the CB and MT heuristics for my NA heuristic.

4. Is there anything you are surprised by or learned from the experiment?

I was kind of surprised the shuffle length had diminishing effects on the nodes explored/time taken to find an optimal solution. For example, there was shuffle size = 40 for MT that took longer than problems with shuffle sizes of 50. I was also surprised that the admissible heuristics outperformed the NA heuristic at lower shuffle levels, yet at higher shuffle levels NA was far quicker in finding a solution than the admissible heuristics, even if it didn't always find an optimal solution.