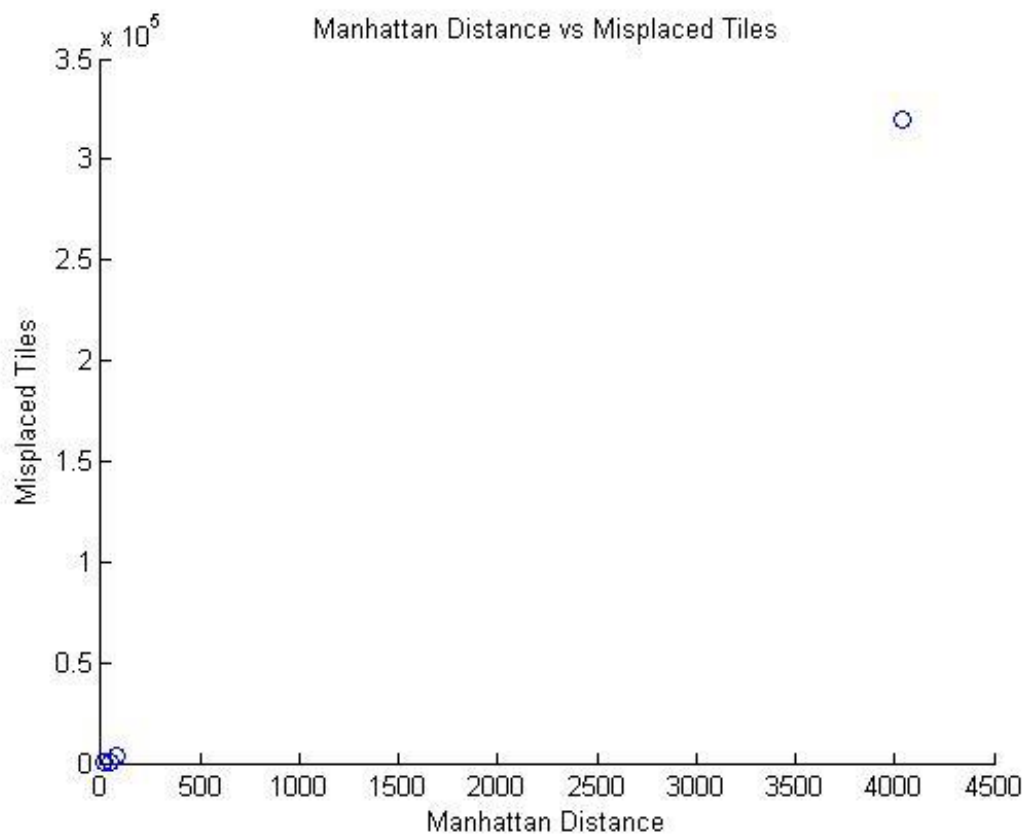


Q1 questions

- a) Which heuristic performs better, Misplaced Tiles or Manhattan distance?

From Graph 1, the Manhattan distance performs better than Misplaced Tiles on average.

- b) Graph of number of nodes explored by the Manhattan distance on x-axis, number of nodes explored by the misplaced tiles heuristic on y-axis. As shown in the graph, as the Manhattan distance explore more nodes, the number of nodes explored by misplaced tiles increases at a higher rate.



Graph 1: Plot of Manhattan Distance against Misplaced Tiles on number of nodes explored

Q2 questions

- a) For each of two specified heuristics admissible, say if it is admissible or not.
- a. Sum Heuristic
 - i. Admissibility (≥ 1 package): Sum Heuristic is not admissible (more states that does not contain optimal Max will have larger sum) .
 - ii. Admissibility (1 package only): Sum Heuristic is admissible (behaves the same as max for the case of 1 package only) .
 - b. Max Heuristic
 - i. Admissibility (≥ 1 package): Max Heuristic is admissible (successor will be the max) .
 - ii. Admissibility (1 package only): Max Heuristic is admissible (max of 1 element is the element itself, therefore successor will be the max) .
- b) Does the information from location map have to be included in the hashable state?

The information from the location map does not need to be included in the hashable state as the current set of carriedJobs, current set of jobsToVisit, currentLocation of courier, and currentTime together uniquely defines the state that will determine the next output direction (note: carriedJobs by itself uniquely define the amount of weight currently on the courier, we do not have to include weight in). Therefore, these can uniquely determine the next successor that will rely on the location map for computation but the location map itself does not need to be in the hashable state as each test will have the exact same location map information.

- c) If part of the state can be computed from other parts, does it have to be part of the hashable state?

If part of the state can be computed from other parts, it does not need to be part of the hashable state (e.g. weight computed from the set of carriedJobs).

- d)
- a. Does stopping the courier from performing a pickup actions at the dropoff location for a job they are currently carrying affect: solution exists? Cost of optimal solution?

No, stopping a courier does not affect whether or not a solution exists and it does not affect the cost of the optimal solution as it is always better to drop off what you currently have to maximize earnings on that dropped item as well as minimize weight carried to be able to pick up current object on that location.

- b. Does blocking these pickup actions make search more efficient?

Yes, blocking these pickup actions make the search more efficient as you can always find a solution that performs as well or better by dropping off before picking up, which allows you to prune these picking up before dropping off nodes away.

- e) If courier is at some location and there is an unstarted job that can be picked up, is it always best for the courier to pickup that job?

No, it is not always best for the courier to pickup that job as that job may be very heavy with a drop off location that takes a long time to travel too which only gives a small amount of earning upon reaching there. It might be better to go to a different location first to gather lighter weight items that requires travelling a shorter time for drop off to earn a larger earning that would otherwise not be earned if the courier picked up that heavy unstarted job that was too heavy in the first place.

- f) Including the current time in the state limits the effectiveness of cycle checking. What problem specific rule can we use to prune states from OPEN list that will still allow us to find maximum revenue solution?

A problem specific rule that allows us to prune states from OPEN list is to consider a state a cycle if the current time is later than a current time of an existing 'similar state' (same location, carrying same jobs, earned same money, and same set of unstarted jobs) as we know it is always better to only execute this exact same 'similar state' with an earlier current time since we can assume that the payoffs always decrease as the delivery times are later.