Problem Formulation and Design

## Problem’s environment properties and Performance measure

## Problem formulation

## Relaxed Problem Technique to define an admissible and consistent heuristic

## Search Method

We have different search methods in which we can trust to solve this problem such as the blind and the informed algorithms. The local search algorithms are not valid here because we are interested in the path of the solution, not the solution itself. Lets classify them in two groups:

### Blind search

#### Breadth-First

It is a valid algorithm because it is Complete and is Optimal, because all the actions have the same cost.

#### Uniform-Cost

This method will act as Breadth-First because all the same actions have the same cost.

#### Depth-First

If we consider that it could be a grid of 2x2 that it just have X (loop) this algorithm is not valid because is not complete and not Optimal. The completeness can be solved if we keep in memory a list of repeated states, but it will be still not optimal.

#### Depth-Limited

Considering that we will use an l < d, this method will not ensure the completeness, because the solution can be in a depth between l +1 and d.

#### Iterative-Deeping

Iterative-Deeping is a good choose in case that we use a blind search because it ensures the completeness and the optimality (all the action have the same cost).

As we have saw, there are some methods such as Breadth-First, Uniform-Cost and Iterative-Deeping that are valid for this problem, but as we have observed in previous questions, the problem is fully observable and also, it is an offline problem, because the environment is static and the board does not change from the initial state, so it will be better to use an informed search algorithm that makes use of the heuristic obtained in the previous question.

### Informed Search

#### Greedy Best First –Search (Based on GraphSearch)

It will make use of the heuristic defined (Manhattan Distance) as the evaluation function in order to sort the frontier, so it is expected to have a better performance than any blind search algorithm.

#### A\*

This algorithm is one of the best algorithms to solve this problems because the evaluation function will use also a function that calculates the actual cost of the path that is being covered ( g(n) ). The function just have to count the number of movement done to calculate g(n) and the add them to h(n) to obtain our evaluation function f(n)

However, this method is not implemented in the jar file provided in the subject, so we will make use of the Best-First search algorithm provided in the practice, which is based in an algorithm that only uses a Heuristic as Evaluation Function.