2022-2023 Fall CmpE480 HW1

Karahan Sarıtaş - 2018400174

- I have implemented all the search algorithms listed below:
  - `BFS`, `DFS`, `UCS`, `GS`, `A*`, `A*2`
- I've implemented A* algorithm with my own heuristic function. Let me explain my heuristic, why it is consistent and admissable in the following section.

Heuristic Function

My heuristic function $h_2(n)$ (n being the current state) = **minimum** number of horizontal or vertical lines that cover all the pegs on the board $-1.$ (I add the -1 so that it becomes equal to zero when there is only one peg left on the board). A line covers the whole row or column on which it is drawn.

Now, the question is, how can we calculate the $h_2(n)$ for each state n? As a result of my research, I found out that this problem can be viewed as finding a **minimum cardinality vertex cover** of a bipartite graph. It turns out that this problem is NP-hard for non-bipartite graphs (ref), meaning that it cannot be solved by a polynomial time algorithm if $P \neq NP$. However, luckily it can be solved for bipartite graphs in polynomial time. I put the reference to the algorithm below.

Consider our board as the adjacency matrix of a *bipartite* graph. There will be a vertex for every row and a vertex for every column. For every cell occupied by a peg, there is an edge between the corresponding row and column vertices. Such a graph is called *bipartite* because there are two types of vertices and edges exist only between vertices of different types. Finding the minimum cardinality vertex cover for bipartite graphs can be done using the Kőnig's theorem in polynomial time.*

*: I used the algorithm provided here: https://tryalgo.org/en/matching/2016/08/05/konig/.

Consistency and Admissibility

We know that every consistent heuristic is admissable. Let's prove that our heuristic is **consistent**. We have to show that the following equation always holds:

$$h_2(n) \leq h_2(n\text{'}) + c(n, a, n\text{'})$$

n stands for the previous state, n' denotes the new state and a stands for the action that takes us from state n to state n'. c is the cost of that action.

1. By definition, $h_2(n)$ gives the minimum number of horizontal or vertical lines to cover all pegs on the board. Let S denote the set of **covering-lines** chosen by our **bipartite minimum cortex cover** algorithm**.**
2. Let's suppose that $h_2(n) > h_2(n\text{'}) + c(n, a, n\text{'})$ for some n, n' and a. Then we can derive: $h_2(n) > h_2(n\text{'}) + c(n, a, n\text{'}) \geq h_2(n\text{'}) + 1$ since the least-cost move costs us 1. Inequality is reduced to $h_2(n) > h_2(n\text{'}) + 1$.
3. In any action a, we can move a peg either along a vertical path or a horizontal path. Let's assume that our peg P jumps *up* along a column, covered by the **covering-line** R, therefore $c(n, a, n\text{'}) = 1$. We have chosen the move *up* intentionally as it is the move with the least cost. If we show that the inequality at (2) cannot hold for the move *up*, then automatically it cannot hold for other moves as well.
4. In each action we can move only one peg either along a row or a column. Therefore performing an action can at most reduce the heuristic h by 1. Possible scenarios are listed below:
   a. By jumping *up*, P cannot eliminate any peg on different columns since it is only allowed to move along the same column. Therefore it cannot reduce any **covering-line** on any other column other than its own column. Same reasoning applies for other moves as well.
   b. By jumping *up*, P cannot eliminate any **covering-line** on a row. Let's suppose that the move of P eliminates a **covering-line** L that covers a row. Since P cannot eliminate any other peg on the same row, it shows that the there was only one peg covered by the **covering-line** L. We also know that, that specific peg was covered by **covering-line** R as well since both P and the eliminated peg were present in the same column. This shows that we can remove **covering-line** L from the set S and still cover all the pegs on the board.

However, it contradicts with the fact that S corresponds to the set of **covering-lines** specified by **bipartite minimum cortex cover** algorithm**.**

5. As shown above, in both of the scenarios, move of P cannot reduce the minimum number of lines more than 1. If our action was not *jump up*, but something else that costs more, then the same reasoning applies. So $h_2(n) \ngtr h_2(n') + 1$, therefore our assumption at step (2) leads to a contradiction regardless of our choice of move.
6. Therefore $h_2(n) \leq h_2(n') + 1 \leq h_2(n') + c(n, a, n') \Rightarrow h_2(n) \leq h_2(n') + c(n, a, n')$

We proved that our heuristic h_2 is **consistent**. It implies that h_2 is also **admissable**.

- To compare the performances of A^* and A_2^*, you can use more sophisticated input boards like those listed below:

```
input1 = ".....\n.atb.\nn...g\nn...a\nh...b\nx...c"
input2 = "...abc..\n..dh....\n..cg....\n..bf....\n..ae....\n........"
```