ISTANBUL TECHNICAL UNIVERSITY COMPUTER ENGINEERING

Assignment 1

Artificial Intelligence (BLG 521E)

Nov 16th, 2021

Ahmet Kemal Yetkin
504201506

Problem 1 - Constructing Admissible Heuristics

1. Construct an admissible heuristic for the given problem, and call it h_1.

Admissible Heuristic: The estimated cost should usually be decrease than or identical to the real value of accomplishing the aim state. A heuristic function is stated to be admissible if it by no means overestimates the value of accomplishing the aim. ($h(n) \le h(n)^*$)

Let's check it for the given example.

- *$h(n)^*$ - the real cost from current state to goal state*

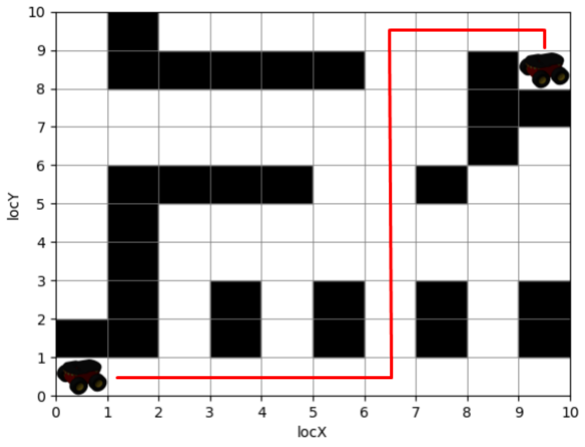


Figure 1. The real cost from current state to goal state

If both robots move on the red line to meet in the single cell, $h(n)^*$ will be 19.

- *$h_1(n)$ – the total Manhattan distance.*

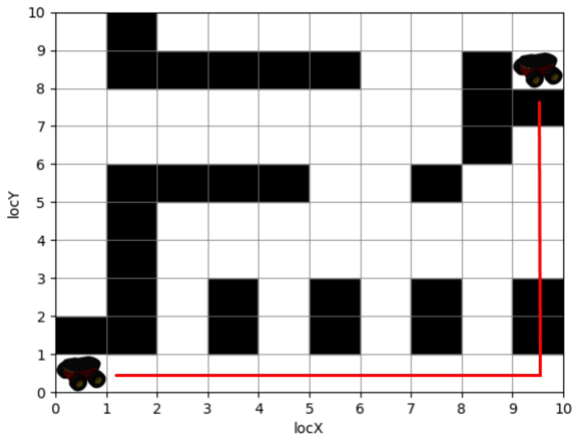


Figure 2. The total Manhattan distance

- If both robots ignore the obstacles and move on the red line, $h_1(n) = 17$

Proof: Since $h_1(n) \le h(n)^*$, the $h_1(n)$ is admissible heuristic.

2. **It seems that Cihan already has his own admissible heuristic function h_2 for the problem, show that h_3 = max (h_1, h_2) is admissible so you can use h_3 for search.**

It's known that h_1 is an admissible heuristic from the previous question. If Cihan's admissible heuristic function $h_2 \geq h_1$, then we can say that h_2 dominates h_1 and also it is better for search.

- h_3 = max (h_1, h_2) – Since h_3 is dominates the h_1 and h_2, it is also admissible heuristic and better for search.

3. **Upon further inspection you see h_1 and h_2 are also consistent heuristic functions, show that max (h_1, h_2) is also a consistent heuristic function.**

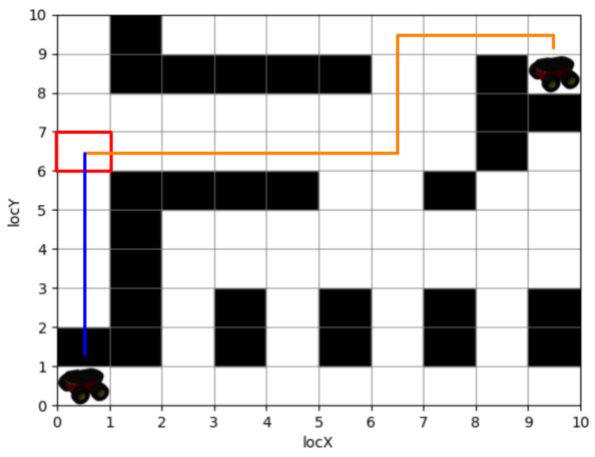
First, let's see how do we know if a heuristic function is consistent.

$$\boldsymbol{A\ heuristic\ is\ consistent\ if}$$

$$\boldsymbol{h(n) \leq c(n, a, n') + h(n')}$$

Let's choose one cell on the maze and mark it with red color for h_1. The orange line will be calculated on real cost for the first robot. The blue line will be calculated on heuristic function for the second robot.

In this situation; the orange line ($c(n, a, n')$) = 13 and the blue line (h(n')) = 6.



As we know from the first question, if both robots move to the single cell the real cost $h(n)^*$ is 19.

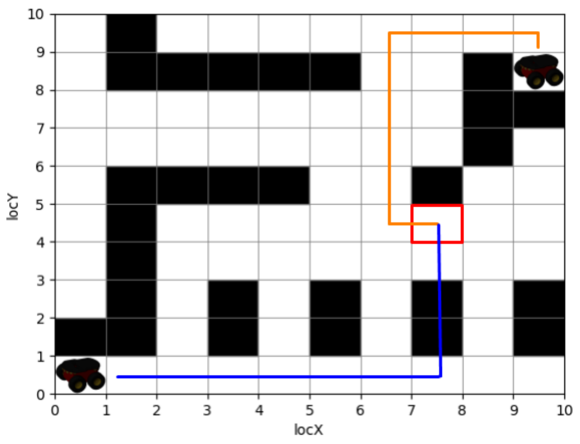
Therefore,

$$\boldsymbol{19 \leq 13\ +\ 6}$$
$$\boldsymbol{19\ \leq\ 19}$$

This shows that the <u>h_1 is consistent heuristic.</u>

Again, let's choose another cell for h_2.

In this situation; the orange line ($c(n, a, n')$) = 10 and the blue line (h(n')) = 11.



$$19 \leq 10 + 11$$
$$19 \leq 21$$

This shows that the h_2 is consistent heuristic.

- Since h_3 is dominates the h_1 and h_2 , h_3 is also consistent heuristic.

While all consistent heuristics are admissible, not all admissible heuristics are consistent.

Problem 2 - Application of Methods in Problem Domains

Topic: NPC Planning

Example case: An Algorithm Applying to NPC Path Planning of Web Games [1]

1. A* search algorithm is a graph traversal and path search algorithm, which is often used in many fields of computer science due to its completeness, optimality, and optimal efficiency. One of the biggest usage areas of the algorithm is games. In games, path planning is made for NPC (Non-Player Character) and it is expected to behave accordingly. Path planning is the core issues in the artificial intelligence field of games, and how to establish an effective method of path planning is still focused on.

   Their algorithm based on both the benefits of global path planner methods and local path planner methods and those of A* algorithm and improved artificial potential field method is proposed for NPC path planning in web games. A* algorithm which only applies to static environments and produces bad effects in dynamic environments is the most representative one of global path planner methods. Artificial potential field method represents local path planner methods best as it has simple structures, less computations and good ability of avoiding obstacle but there are problems about local minimum. Their new method combined both global path planner methods and local path planner methods is proposed in this paper. Based on the motion strategy of NPC and the feature of using grid maps in web games, A* algorithm and improved artificial potential field method are combined in this new method.

2. Path planning of NPC can be described as locating the optimal path from the starting point to the destination for NPC. Ideal result isn't achieved by applying the traditional path planner for NPC in the dynamic circumstances.

A Algorithm* is explained in the third question. However, briefly, A* search algorithm is a typical and heuristic search algorithm that has been often applied to mobile computing of NPC or BOT of online games. Iits heuristic formulated function:

$$f(n) = g(n) + h(n) .$$

Artificial potential field method is frequently used in local path planner, the principle of which is that NPC is regarded as the particle its movement controlled by resultant forces which is composed of repulsive force from obstacles and attractive force from the object point.

The paper proposes a new algorithm based on both the benefits of global and local path planner and those of A star algorithm and improved artificial potential field method and the train of thought as follows. **You can see my pseudo code below. It is not exactly representing pseudo-code; however, it gives an idea for the algorithm. It shows the how this method is used to solve problem.**

```
while (arrive(object_point) = False):
    According to known environmental information, A star algorithm has
    found an utilization in global path planner, created and optimized
    node sequence of path.

    while (get(object_point) = False):
        Take the adjacent to current path node as sub-object point.

    Get the NPC position information and local environmental
    information within its field of vision. Compute the Fatt attractive
    force, the Frep repulsive force and the F resultant force.
    According to the direction of the resultant force and the velocity
    of NPC the new position isn't computed until that NPC arrives at
    the sub-object point.

    When NPC is oscillated or stagnated, the position of NPC, here, is
    taken as the starting point. A* algorithm is again applied on the
    path planning to compute and optimize the path between the starting
    point and the sub-object point. Then the optimized path will
    replace the original one and get to the sub-object point.

P.S.: in every while loop node count will be increased.
```

According to the conclusion of the paper, the path planned by this algorithm is shorter, more real, more smoothed and the obstacles can be effectively avoided when NPC moves, which promotes NPC intelligent performance.

Here is the simulation of the solution. You can see the NPC follows a straight path when there is no obstacle.

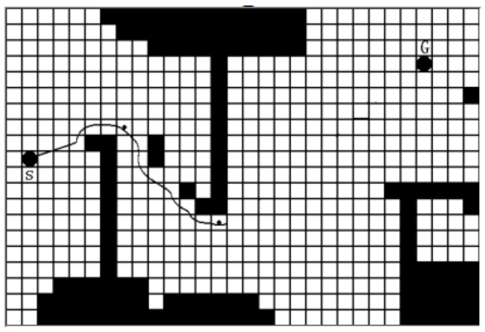


Fig. 3 The process of simulations results

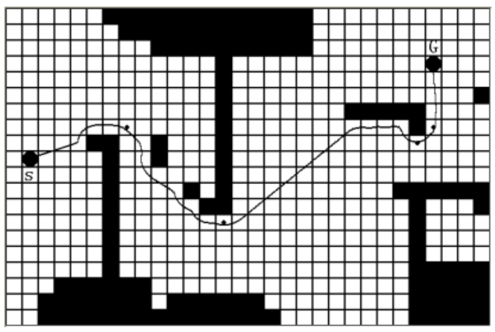


Fig. 4 Simulations results

3. **Reference:** Zhang Cheng, Zhang Kun, Zhang Yingqian, and He Dan: An Algorithm Applying To NPC Path Planning Of Web Games (2014)

Problem 3 - Problem Solving with Search Algorithms

a) **Formulate this problem in a well-defined form. Explain your state and action representations in detail. If you have used different representations in your implementation, explain them all.**

In this problem, Pac-Man agent will find paths through the grid world environment, to reach a particular location of apple. There are state and some actions:

State: The apple and the character positions.
Action: The character movement. The character can only move from 1 grid rectangle to another at a single time step. Also, it can move in 4 directions: right, up, left, down.

If the agent encounter with a wall, it will move any other direction.

b)

		A*
LEVEL 3	Number of nodes generated	175
	Number of expanded nodes	87
	Max. number of nodes kept in memory	18
	Elapsed solve time	0.006981
	Elapsed time step	17

c) D* star lite will be run much more faster than A* algorithm. However, it may be change for small clusters.