Group Assignment 1
Artificial Intelligence

Liquid Container Problem

-

-

-

-

Ni Hao Group 1:

Izaaz Rahman Akbar (21/472855/PA/20348)

Matthew Tan (21/478240/PA/20736)

Rabbani Nur Kumoro (21/472599/PA/20310)

William Hilmy Susatyo (21/472585/PA/20308)

1. Problem

Artificial Intelligence is the study of how to make computers function or perform as well as humans or even better in solving complex tasks **[1]**. The problem that we attempt to solve is called the Liquid Container Problem. The main objective of the Liquid Container Problem is to measure the amount of liquid in a container without any measurement device by completely filling and pouring liquid to and from at least two containers of known volume.

In general, this problem necessitates the conversion of one predicted state to another through several operating methods **[2]**. In practice, there may seem to be no solution to this problem, or there may be more than one answer. There are various methods to tackle these challenges at times, but we will put our main attention on implementing the Admissible Heuristic and A* Search Algorithm.

For example, there are two containers, let's assume that container_1 is "a" and container_2 with "b" and the volume should be $0 < a < b$. Both containers will initially be empty, and they don't have a mark to measure small quantities. Now, we need to measure the target liters "d" of liquid by using these two containers in a condition where $d < b$. For this problem, we will use the following three operations to measure the target quantities:

1. Empty the Container.
2. Fill a Container.
3. Pour the Liquid of one Container into another one until one of them is either full or empty.

The underlying logic of this problem is how the steps are carried out to obtain a volume of d liters of liquid by utilizing the available container. In simple terms, the steps that can be taken to solve the problem are only filling in and excreting liquid out of the container. In theory, it's quite simple, but in reality, many users are constrained by the logic of the steps that must be taken, especially if the liquid container used is more than two with various volumes of liquid.

For demonstrating the methods to solve this problem using the A* Algorithm, let's assume that "a" can hold 8 liters while " b can hold" 6 liters. Consequently, there will be a faucet that can supply the liquid infinitely. The initial conditions of all the containers are empty. The problem to be solved is to make one of the containers hold 4 liters of liquid.

Now we will break down the problem into simpler steps. First, we would symbolize “a” and “b” as (a, b) where “a” can contain any amount of liters from 0 to 8 and b the same rules from 0 to 6. Next, we should note that the initial value of both containers is (0,0) and the final objective to be obtained is either (a, 4) or (4, b). Then, we should establish a series of production rules to describe a tree structure for the A* Algorithm. (0,0) will be the initial node of the tree and the series of rules in the table below will help us trace all possible leaf nodes. The tracing will stop until all of the rules are being carried out by all of the nodes. The answer will be found if it completely defines all of the production processes.

Production Process Table		
No	Production Rules	Information
1.	a < 8	(8, b). Fill the liquid container “a”.
2.	b < 6	(x, 6). Fill the liquid container “b”.
3.	a > 0	(0, b). The contents of the liquid container “a” are drained until empty.
4.	b > 0	0 (a, 0). The contents of the liquid container “b” are drained until empty.
5.	a + b >= 8 && b > 0	(8, b-(8-a)). The contents of the liquid in container “b”: are poured into the full container “a”.
6.	a + b >= 6 && a > 0	(a-(6-b),b). The liquid in container “a” is poured into container “b” until it's full.
7.	a + b <= 8 && b > 0	(a+b, 0). All of the liquid that is in the container “b” is poured into the container “a”.
8.	a + b <= 6 && a > 0	(0, a+b). All of the liquid in container “a” is poured into container “b”.

2. Admissible Heuristic

A particular distance is considered an admissible heuristic as long as the estimation of the heuristic function never overestimates the actual costs.

For this problem, we provide 4 different admissible heuristic functions.

1. General Function

The heuristic function proposed for a liquid container problem, $h(x, y) = (ax) + (by)$, and has the lowest value at (0,0), because if we input (0,0), the outcome is 0. The x and y represent the current state of the containers as in how much liquid is in a container. The a and b represent the volume of each container.

This, however, is not a good heuristic function that we want to use because as we reach the goal node, which is either (4,b) or (a,4), the output of the function is not 0, but instead, the initial state will output 0. In making a heuristic function, it is better to make it such that if we input the goal node, the heuristic value is 0, and thus indicates that the node is the goal.

2. Manhattan Distance

Manhattan Distance is determined by computing the total number of distances during horizontal and vertical movement in order to reach the target point from the initial point. Thus, it can be concluded that Manhattan Distance is constructed by combinations of straight movements.

Therefore, the formula of Manhattan Distance is defined as below,

$$m(n) = |x_n - x_{goal}| + |y_n - y_{goal}|$$

where x_n and y_n consecutively denotes the $x - coordinate$ and $y - coordinate$ of the initial point. Moreover, x_{goal} and y_{goal} represent the $x - coordinate$ and $y - coordinate$ of the destination point. The proof that Manhattan Distance is considered an admissible heuristic will be explained in the following paragraph, According to the definition of admissible heuristic, we will first prove that

$$m(n) \leq h^*(n)$$

By the definition of a heuristic function, $h*(n) = 0$ in the case of n* is the goal state.

Assume that n* < $m(n_0)$ for the initial state n_0. Since each movement can only results in in one particular position, performing a movement can reduce $m(n)$ by one. Based on the fact that the goal can be reached in n* movement, we obtain $h^*(n) \geq m(n) - n* > 0$, which leads to a contradiction since $h^*(n)$ should be zero. Hence, eventually implies $m(n_0) \leq n*$ for all n_0, which results $m(n) \leq h*(n)$, indicating that $m(n)$ is admissible. (Q.E.D)

3. Euclidean Distance

In general, Euclidean Distance calculates the distance between two certain points by finding the length of a straight line or segment that connects those points. The disadvantage of Euclidean Distance is that it is more complicated than Manhattan Distance due to the larger area that needs to be explored in order to find a path. In addition, the Euclidean Algorithm is also prone to what is called the curse of dimensionality, a condition that occurs when analyzing data in high-dimensional spaces.

The formula of Euclidean Distance itself could be represented as below,

$$e(n) = \sqrt{(x_n - x_{goal})^2 + (y_n - y_{goal})^2}$$

Thus, it can be clearly observed that Euclidean Distance is simply the square root of the sum of squared distances between corresponding coordinates of two particular points. Observe that the formula is similar to the Pythagorean Theorem Formula.

However, the proof that Euclidean Distance is an admissible heuristic is quite trivial. We will first prove that the Euclidean Distance is always less than the Manhattan Distance.

Observe that,

$$e(n)^2 = (x_n - x_{goal})^2 + (y_n - y_{goal})^2$$

That'd imply,

$$(x_n - x_{goal})^2 + (y_n - y_{goal})^2 \leq (x_n - x_{goal})^2 + 2(x_n - x_{goal})(y_n - y_{goal}) + (y_n - y_{goal})^2$$

Also observe that,

$$(x_n - x_{goal})^2 + 2(x_n - x_{goal})(y_n - y_{goal}) + (y_n - y_{goal})^2 = m(n)^2$$

That'd give,

$$e(n)^2 \leq m(n)^2 \Leftrightarrow e(n) \leq m(n)$$

Hence, it can be deduced that the euclidean distance is less or equal than the Manhattan distance. Since The Manhattan Distance is considered an admissible heuristic, Euclidean Distance should be an admissible heuristic too (Q.E.D)

4. Chebyshev Distance

Chebyshev Distance is widely known as L_∞ $metric$, obtain the distance from the initial point to the destination point by determining the maximum difference between the destination and initial point in the coordinate dimension.

The general formula of Chebyshev Distance is as below,

$$c(n) = max\{|x_n - x_{goal}|, |y_n - y_{goal}|\}$$

According to the formula above, we might notice that this heuristic function is merely the greatest absolute distance in one dimension given two N-dimensional points.

On the other hand, since it is also known as L_∞ $metric$, the formula of Chebyshev Distance could be described as,

$$c(n) = \lim_{p \to \infty} \sqrt[p]{(x_n - x_{goal})^p + (y_n - y_{goal})^p}$$

However, without loss of generalization, by assuming the value of $(x_n - x_{goal})$ is greater than $(y_n - y_{goal})$, notice that,

$$\lim_{p \to \infty} \sqrt[p]{(x_{n} - x_{goal})^{p} + (y_{n} - y_{goal})^{p}} \geq \lim_{p \to \infty} \sqrt[p]{(x_{n} - x_{goal})^{p}}$$

$$= \lim_{p \to \infty} (x_{n} - x_{goal})$$

$$= (x_{n} - x_{goal})$$

$$= max\,\{|x_{n} - x_{goal}|,\ |y_{n} - y_{goal}|\}$$

On the other hand,

$$\lim_{p \to \infty} \sqrt[p]{(x_{n} - x_{goal})^{p} + (y_{n} - y_{goal})^{p}} \leq \lim_{p \to \infty} \sqrt[p]{2(x_{n} - x_{goal})^{p}}$$

$$= (x_{n} - x_{goal}) \lim_{p \to \infty} 2^{\frac{1}{p}}$$

$$= (x_{n} - x_{goal})$$

$$= max\,\{|x_{n} - x_{goal}|,\ |y_{n} - y_{goal}|\}$$

Suppose $max\,\{|x_{n} - x_{goal}|,\ |y_{n} - y_{goal}|\}$ = p

Since

$$p \leq \lim_{p \to \infty} \sqrt[p]{(x_{n} - x_{goal})^{p} + (y_{n} - y_{goal})^{p}} \leq p$$

It can be deduced that,

$$\lim_{p \to \infty} \sqrt[p]{(x_{n} - x_{goal})^{p} + (y_{n} - y_{goal})^{p}} = max\,\{|x_{n} - x_{goal}|,\ |y_{n} - y_{goal}|\}$$

Furthermore, we will provide the proof that Chebyshev Distance is considered as an admissible heuristic, which is relatively trivial.

Observe that,

$$max\, \{|x_n - x_{goal}|\, ,\ |y_n - y_{goal}|\} \ \leq\ |x_n - x_{goal}| \ +\ |y_n - y_{goal}|$$

which implies that Chebyshev Distance is less than Manhattan Distance. Since we know that Manhattan Distance is admissible, it can be inferred that Chebyshev Distance is also an admissible heuristic. (Q.E.D)

3. A* Example and Explanation

A* Search Algorithm is considered one of the well-known methods that are widely used in graph traversal and pathfinding **[3]**. It is a Search Algorithm that functions to perform input analysis, evaluate a number that could be a solution, and produce a solution. So that the route can be found from the initial node to the destination node, a search is carried out using a graph.

Moreover, the A* can also be defined as the extension of the Dijkstra Algorithm. Mainly, the A* could be represented in a function $f(k)$ that approximates the total cost of the path given several particular nodes. The formula of $f(k)$ itself is defined as below:

$$f(k) \ = \ g(k) \ + \ h(k)$$

where,

$f(n) \ = \ total\ approximate\ cost\ of\ path\ through\ \ node\ k$

$g(n) \ = \ cost\ so\ far\ to\ reach\ certain\ node\ k$

$h(n) \ = \ approximate\ cost\ from\ node\ k\ to\ goal's\ node$

The working principle of this A* Algorithm is to run a transversal on each node one by one to get the closest path. A* Algorithm will calculate the distance of one of the paths, and then the path is saved, and then calculate the distance of the other path. If the calculation for all paths has been completed, then the closest path will be chosen by A* Algorithm **[4]**. The work process flow diagram of the A* Algorithm is shown in Figure 3.

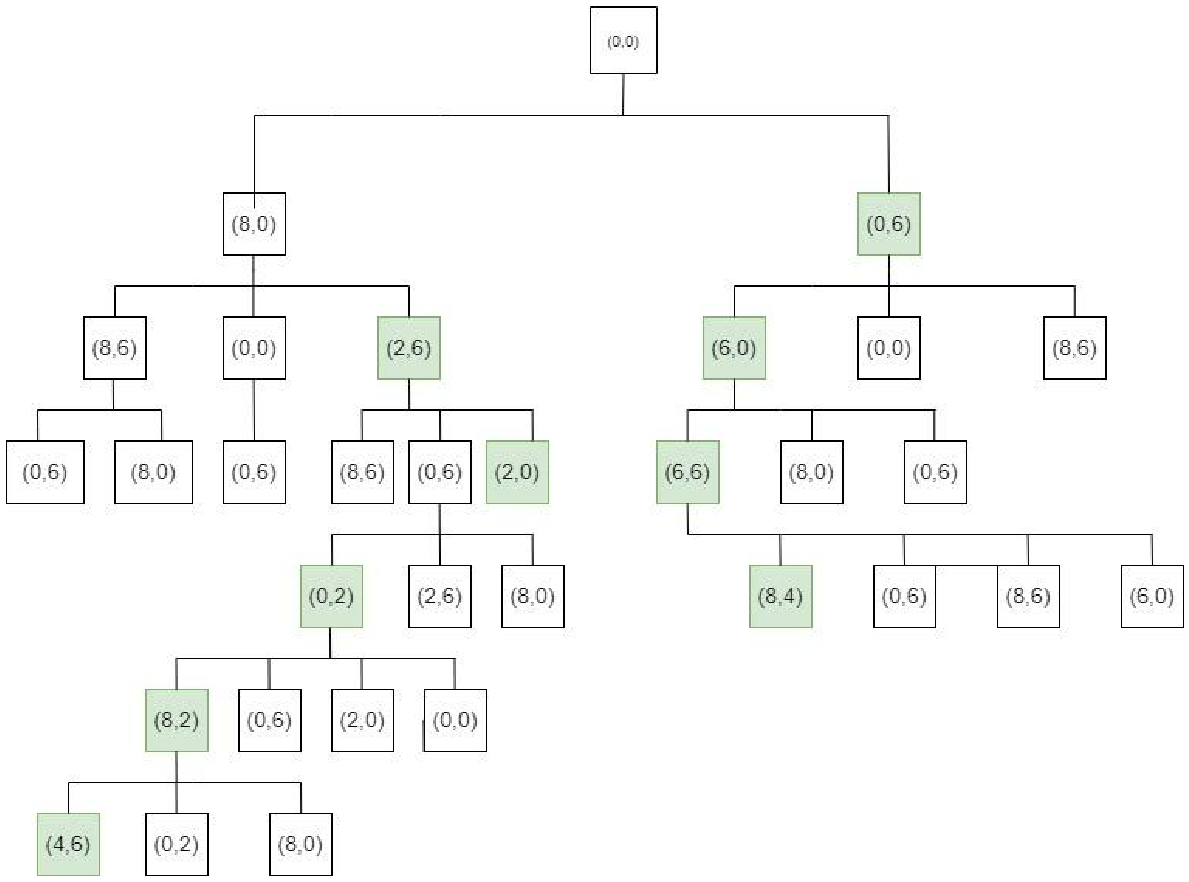

Figure 3. A* Algorithm Binary Tree Search

From the binary tree structure of the A* Algorithm above, it can be determined 4 optimal steps for the solution, namely: (0-0),(0-6),(6-0),(6-6),(8-4).

REFERENCES

[1] Wijaya, E., "*Analysis of the Use of the Breadth First Search Algorithm in the Concept of Artificial Intelligence*", TIME Journal, Vol. II No 2, p. 18-26, 2013, ISSN : 2337 – 3601

[2] Harianja, F., "*Application of the A* Algorithm to Optimization Problems for Dynamic Water Solution Search*". Pelita Informatika Journal, Vol. IV, No. 3, August, 2013, ISSN : 2301-9425

[3] Setiawan, K., et al,. "*Calculating the Shortest Route Using A* Algorithm with Euclidean Distance Function*". National Seminar on Information and Communication Technology (SENTIKA). Yogyakarta. 2018

[4] Hermanto, D.; Dermawan, S., "*Application of the A-Star Algorithm as the Shortest Route Finder on the Hexapod Robot*". National Journal of Electrical Engineering, Vol. 7, No. 2, p-ISSN: 2302-2949, e-ISSN: 2407 – 7267. 2018