*<Assignment 1, Machines and Intelligence COM1005>*

*Registration No: 170163831, Name: Junhyeok, Seo*

1. Test results

[In RamblersState]

(1) RamblerState: it has method Coords’ variables and one integer instance for local cost.

(2) getY and getX: each returns cY and cX coordinates

(3) goalP: it returns true for 2 different instances with identical states at the same time, false for 2 instances with different states.

(4) calCost: it returns cost with mathematical calculation.

(5) getSuccessors: it returns succs, and include if statements to store correct local cost and coordinates depends on direction.

(6) sameState: it returns true for 2 different instances with identical states at the same time, false for 2 instances with different states.

(7) RamblerState (in search4): it has method Coords’ variables and two integer instances for local cost and estimates of the remaining cost.

(8) mDistance: it returns dis with mathematical calculation.

(9) eDistance: it returns dis with mathematical calculation.

(10) difHeight: it returns dis with mathematical calculation.

[In RamblersSearch]

(1) getMap: returns map.

(2) getGoalY and getGoalX: each returns goalY and goalX.

(3) RamblersSearch: it has TerrainMap method and Coords method.

2. Experimental Results

**(1) Branch-and-Bound**

I have used start and end points in the Terrian Map of **tmx.pgm,** and assessed the efficiency of branch-and-bound in this domain. The two tables below show that the number of start, end points and efficiency of branch-and-bound in each tmx map and diablo11 map (**diablo11.pgm**).

<tmx map>

|  |  |  |
| --- | --- | --- |
| Start points (y, x) | End points (y, x) | Efficiency |
| (15, 15) | (0, 0) | 0.14351852 |
| (3, 15) | (2, 0) | 0.11797753 |
| (3, 6) | (1, 5) | 0.22222222 |
| (3, 7) | (3, 5) | 0.50000000 |
| (0, 7) | (14, 14) | 0.22222222 |
| (0, 8) | (8, 15) | 0.19277108 |
| (15, 4) | (2, 13) | 0.25000000 |
| (15, 14) | (12, 3) | 0.10948905 |
| (5, 3) | (10, 3) | 0.04519774 |
| (1, 13) | (1, 1) | 0.07894736 |

<diablo11 map>

|  |  |  |
| --- | --- | --- |
| Start points (y, x) | End points (y, x) | Efficiency |
| (15, 15) | (0, 0) | 0.05525846 |
| (3, 15) | (2, 0) | 0.09550562 |
| (3, 6) | (1, 5) | 0.22222222 |
| (3, 7) | (3, 5) | 0.37500000 |
| (0, 7) | (14, 14) | 0.06111111 |
| (0, 8) | (8, 15) | 0.06349207 |
| (15, 4) | (2, 13) | 0.08487084 |
| (15, 14) | (12, 3) | 0.08287293 |
| (5, 3) | (10, 3) | 0.08450704 |
| (1, 13) | (1, 1) | 0.11304348 |

**(2) A\***

I have implemented A\* search for rambler’s problems three times. The three tables below describe that that the number of start, end points and efficiency of A\* in diablo11 map **(diablo11.pgm)** with three different ways to calculate the remaining cost to the goal. Which are the Manhattan distance, the Euclidean distance and the Height difference.

<diablo11 map with the Manhattan distance>

|  |  |  |
| --- | --- | --- |
| Start points (y, x) | End points (y, x) | Efficiency |
| (15, 15) | (0, 0) | 0.16062176 |
| (3, 15) | (2, 0) | 0.47222222 |
| (3, 6) | (1, 5) | 0.50000000 |
| (3, 7) | (3, 5) | 1.00000000 |
| (0, 7) | (14, 14) | 0.07638890 |
| (0, 8) | (8, 15) | 0.08465608 |
| (15, 4) | (2, 13) | 0.13772455 |
| (15, 14) | (12, 3) | 0.62500000 |
| (5, 3) | (10, 3) | 0.16216215 |
| (1, 13) | (1, 1) | 0.68421054 |

<diablo11 map with the Euclidean distance>

|  |  |  |
| --- | --- | --- |
| Start points (y, x) | End points (y, x) | Efficiency |
| (15, 15) | (0, 0) | 0.11654135 |
| (3, 15) | (2, 0) | 0.24285714 |
| (3, 6) | (1, 5) | 0.4000000 |
| (3, 7) | (3, 5) | 0.1000000 |
| (0, 7) | (14, 14) | 0.07142857 |
| (0, 8) | (8, 15) | 0.07920792 |
| (15, 4) | (2, 13) | 0.11330049 |
| (15, 14) | (12, 3) | 0.27272728 |
| (5, 3) | (10, 3) | 0.13043478 |
| (1, 13) | (1, 1) | 0.43333334 |

<diablo11 map with the Height difference>

|  |  |  |
| --- | --- | --- |
| Start points (y, x) | End points (y, x) | Efficiency |
| (15, 15) | (0, 0) | 0.77500000 |
| (3, 15) | (2, 0) | 0.25757575 |
| (3, 6) | (1, 5) | 0.57142860 |
| (3, 7) | (3, 5) | 1.00000000 |
| (0, 7) | (14, 14) | 0.11000000 |
| (0, 8) | (8, 15) | 0.18604651 |
| (15, 4) | (2, 13) | 0.12169312 |
| (15, 14) | (12, 3) | 0.55555560 |
| (5, 3) | (10, 3) | 0.17647060 |
| (1, 13) | (1, 1) | 0.38461540 |

**(3) Is A\* more efficient than branch-and-bound?**

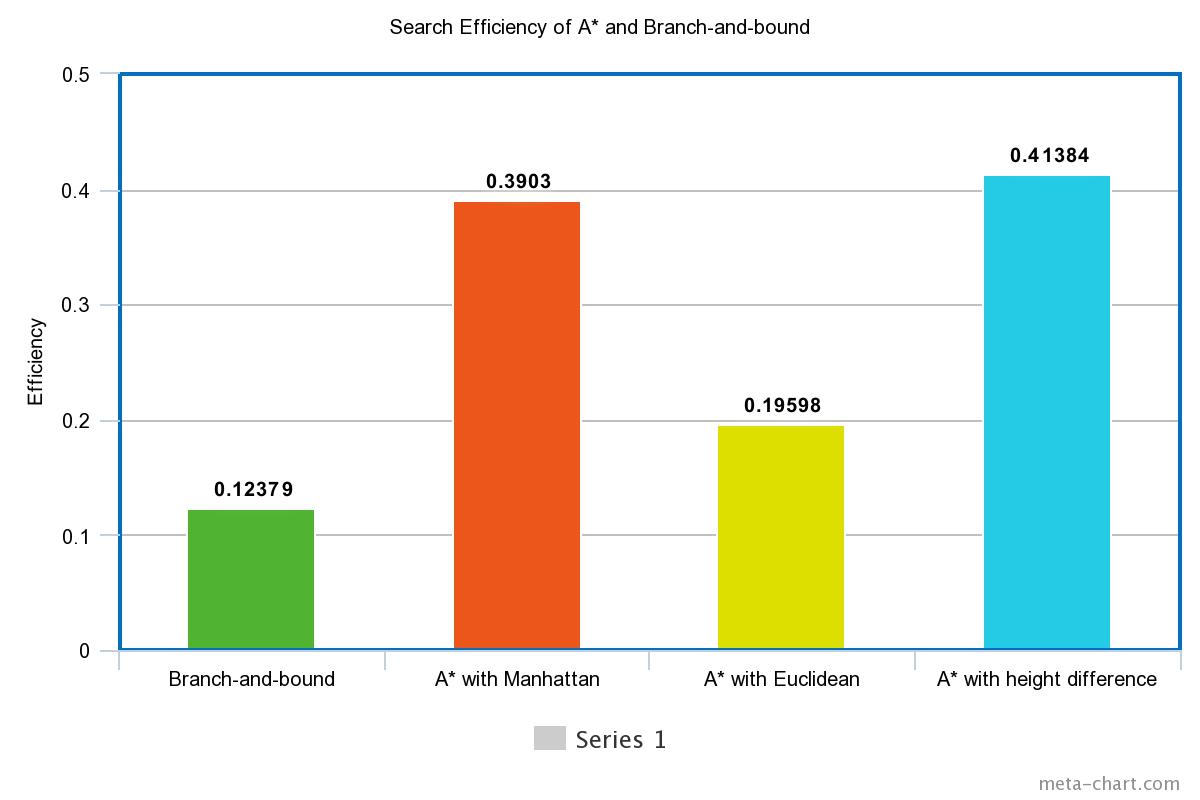
As the results of the efficiency of each A\* and branch-and-bound, the A\* is more efficient. So the hypothesis is true. The two tables below show the result of all and the mean of efficiency of A\* and Branch-and-bound, and the bar graph presents the result to show how A\* is more efficient than branch-and-bound.

<The result >

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| diablo11 map |  |  | Efficiency |  |  |
| Start points  (y, x) | End points  (y, x) | Branch-and-bound | A\* with Manhattan | A\* with Euclidean | A\* with height difference |
| (15, 15) | (0, 0) | 0.05525846 | 0.16062176 | 0.11654135 | 0.77500000 |
| (3, 15) | (2, 0) | 0.09550562 | 0.47222222 | 0.24285714 | 0.25757575 |
| (3, 6) | (1, 5) | 0.22222222 | 0.50000000 | 0.4000000 | 0.57142860 |
| (3, 7) | (3, 5) | 0.37500000 | 1.00000000 | 0.1000000 | 1.00000000 |
| (0, 7) | (14, 14) | 0.06111111 | 0.07638890 | 0.07142857 | 0.11000000 |
| (0, 8) | (8, 15) | 0.06349207 | 0.08465608 | 0.07920792 | 0.18604651 |
| (15, 4) | (2, 13) | 0.08487084 | 0.13772455 | 0.11330049 | 0.12169312 |
| (15, 14) | (12, 3) | 0.08287293 | 0.62500000 | 0.27272728 | 0.55555560 |
| (5, 3) | (10, 3) | 0.08450704 | 0.16216215 | 0.13043478 | 0.17647060 |
| (1, 13) | (1, 1) | 0.11304348 | 0.68421054 | 0.43333334 | 0.38461540 |

<The mean of efficiency of A\* and Branch-and-bound>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Branch-and-bound | A\* with Manhattan | A\* with Euclidean | A\* with height difference |
| Mean | 0.12379 | 0.3903 | 0.19598 | 0.41384 |



(4) why is the best choice of estimates for A\*?

As the bar graph shows, the A\* with the height difference has the highest of efficiency. Hence the best choice could be the A\* with the height difference. However, the difference between A\* with the Manhattan distance and A\* with the height difference is not huge. They are quite similar high efficiency.

And also graph shows that the efficiency of Branch-and-bound is the lowest among of them, which are A\*. So using A\* can give the efficiency gain which helps to estimate more accurately.