Class: CISC6525 Artificial Intelligence

Professor: Dr Damian Lyons

Prepared by: Yuliya Akchurina

Subject: Assignment 1, Out 2/18 - Due 3/4

Python code:

The code was changed in the following way:

Three additional heuristics were created as functions to be compared to straight line distance heuristic. Functions for Manhattan distance, the sum of the straight-line distance and Manhattan distance and the average of Manhattan distance and straight-line distance.

```
# Heuristic 1: Straight Line Distance ((x2-x1)^2 + (y2-y1)^2)^1/2
def sld(city1,city2):
```

```
# Heuristic 2: Manhattan Distance |x2-x1|+|y2-y1|
def md(city1, city2):
```

```
# Heuristic 3: Sum of first two heuristics
def sumd(city1, city2):
```

```
# Heuristic 4:  Average of first two heuristics
def avgd(city1, city2):
```

The 10 city pairs selected:

```
cities_list = [["oradea", "pitesti"], ["zerind" , "craiova"], ["timisoara", "sibiu"],
["sibiu", "lugoj"], ["fagaras", "craiova"], ["drobeta", "oradea"],
["rimnicu", "lugoj"], ["arad", "pitesti"], ["bucharest", "oradea"],
["giurgiu", "timisoara"]]
```

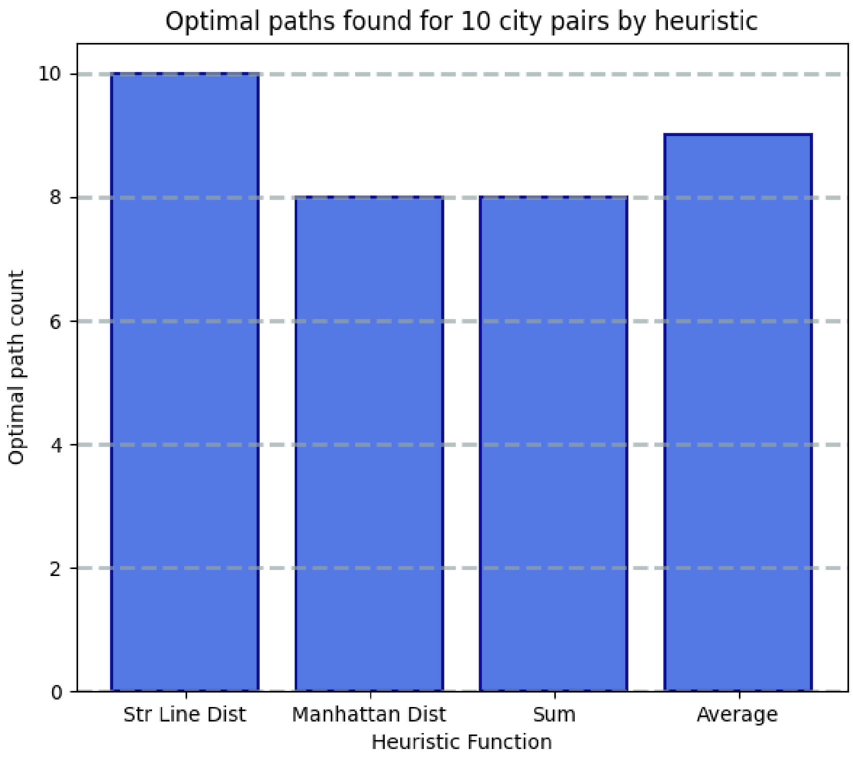
The predetermined optimal (shortest) path between the 10 city pairs:

```
optimal_path = [["oradea", "sibiu", "rimnicu", "pitesti"],
                ["zerind", "arad", "sibiu", "rimnicu", "craiova"],
                ["timisoara", "arad", "sibiu"],
                ["sibiu", "arad", "timisoara", "lugoj"],
                ["fagaras", "sibiu", "rimnicu", "craiova"],
                ["drobeta", "craiova", "rimnicu", "sibiu", "oradea"],
                ["rimnicu", "craiova", "drobeta", "mehadia", "lugoj"],
                ["arad", "sibiu", "rimnicu", "pitesti"],
                ["bucharest", "pitesti", "rimnicu", "sibiu", "oradea"],
                ["giurgiu", "bucharest", "pitesti", "rimnicu", "sibiu", "arad", "timisoara"]]
```

The code has 2 parts. In the part 1 the 4 heuristics were passed as a parameter to the ASTARtreesearch function one after another in a for loop to compute the paths and compare the results to the optimal paths between the 10 city pairs. The results are plotted in a bar chart.

In the part 2 the program is asking for a user input, passes it to the ASTARtreesearch which runs only the corresponding heuristic, and prints out the results of a single heuristic search as paths between 10 city pairs. It also compares the results to the optimal paths and provides the accuracy list and accuracy count.

Plot and conclusions



Above is the plot for the outcomes of the 4 heuristics run on the 10 city pairs compared to the optimal pathways between the 10 city pairs, optimal being the shortest distance.

Comparing the four heuristic formulas we can see that straight-line distance always returns the shortest distance between the two points compared to the Manhattan distance. We can order the heuristics from shortest to longest distance they produce: straight line heuristic (the shortest), Manhattan distance heuristic, average of the to, sum of the two (the longest). Since the A star search is only as good as its heuristic, the results of 10 city pairs searches are consistent with this heuristic distance optimality order.

The A star search cost optimality depends on the heuristic properties. The heuristic must be admissible for the A star search to result in cost optimal outcome.

The straight-line distance heuristic performs the best, resulting in the shortest routes found for all 10 city pairs examples. Straight line distance heuristic is an admissible and consistent heuristic.

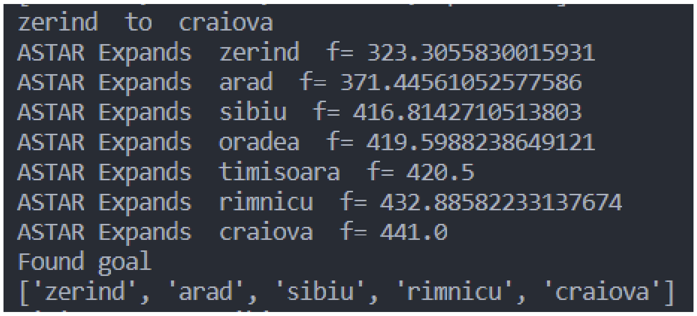
The Manhattan distance heuristic predicts 8 out 10 optimal pathways. According to the definition of admissibility and consistency the Manhattan distance heuristic is not an admissible heuristic and not a consistent heuristic.

Manhattan distance works better for the problems where the points are arranged in the form of the grid and the priority is given to the distance along the grid (city blocks, chess) vs Euclidean distance that always produces the shortest distance regardless of the coordinate system used.

The sum heuristic predicts 8 out of 10 as the shortest routes. This outcome is expected, the sum is not the optimal heuristic since it is not the shortest distance anymore. When using inadmissible heuristic, we allow for the search result to be not optimal. The computational benefit of using the inadmissible heuristic might be that we are opening lower number of nodes while performing the search, which is supported by the search comparison.

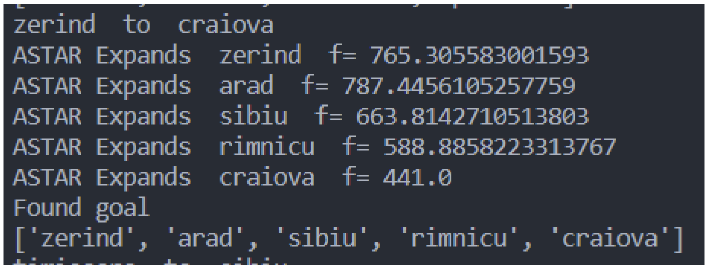
The screenshots of expansions are the same for Manhattan distance heuristic and the Sum of two heuristic.

Straight line distance heuristic from Zerind to Craiova opens 7 nodes



```
zerind  to  craiova
ASTAR Expands  zerind  f= 323.3055830015931
ASTAR Expands  arad  f= 371.44561052577586
ASTAR Expands  sibiu  f= 416.8142710513803
ASTAR Expands  oradea  f= 419.5988238649121
ASTAR Expands  timisoara  f= 420.5
ASTAR Expands  rimnicu  f= 432.88582233137674
ASTAR Expands  craiova  f= 441.0
Found goal
['zerind', 'arad', 'sibiu', 'rimnicu', 'craiova']
```

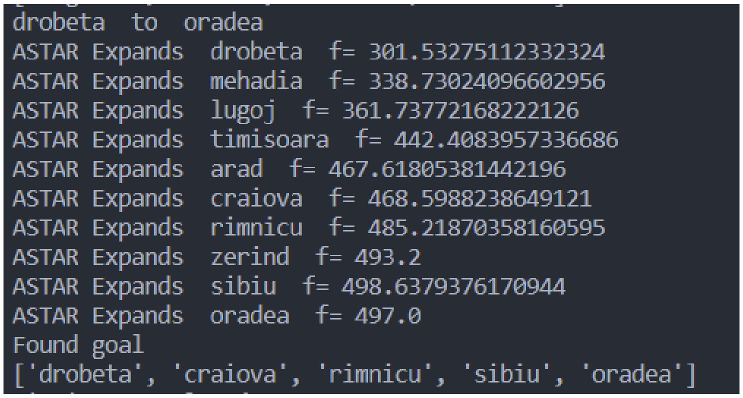
Sum heuristic from Zerind to Craiova opens 5 nodes



```
zerind  to  craiova
ASTAR Expands  zerind  f= 765.305583001593
ASTAR Expands  arad  f= 787.4456105257759
ASTAR Expands  sibiu  f= 663.8142710513803
ASTAR Expands  rimnicu  f= 588.8858223313767
ASTAR Expands  craiova  f= 441.0
Found goal
['zerind', 'arad', 'sibiu', 'rimnicu', 'craiova']
```

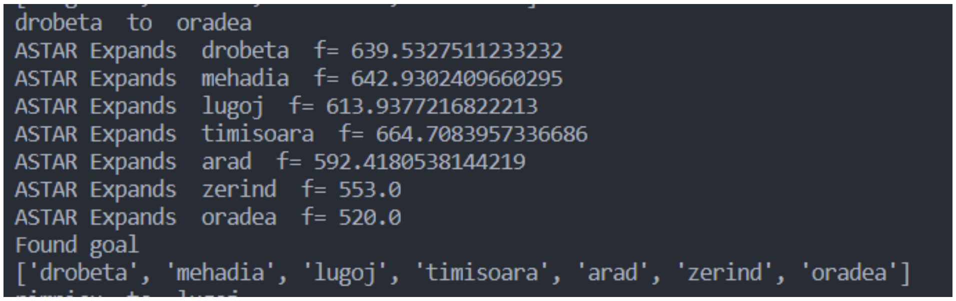
Both result in the same optimal path

Straight line distance heuristic from Drobeta to Oradea opens 10 nodes and returns the optimal path



```
drobeta  to  oradea
ASTAR Expands  drobeta  f= 301.53275112332324
ASTAR Expands  mehadia  f= 338.73024096602956
ASTAR Expands  lugoj  f= 361.73772168222126
ASTAR Expands  timisoara  f= 442.4083957336686
ASTAR Expands  arad  f= 467.61805381442196
ASTAR Expands  craiova  f= 468.5988238649121
ASTAR Expands  rimnicu  f= 485.21870358160595
ASTAR Expands  zerind  f= 493.2
ASTAR Expands  sibiu  f= 498.6379376170944
ASTAR Expands  oradea  f= 497.0
Found goal
['drobeta', 'craiova', 'rimnicu', 'sibiu', 'oradea']
```

Sum heuristic from Drobeta to Oradea opens 7 nodes and the result is not an optimal path.

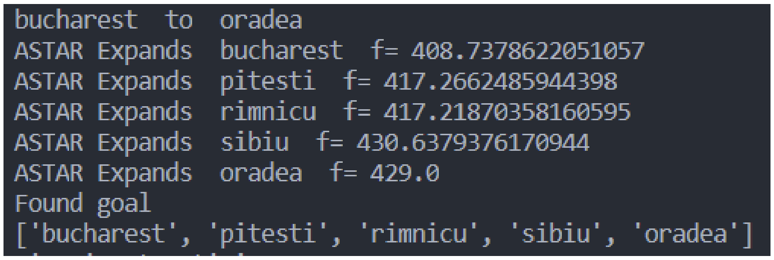


```
drobeta  to  oradea
ASTAR Expands  drobeta  f= 639.5327511233232
ASTAR Expands  mehadia  f= 642.9302409660295
ASTAR Expands  lugoj  f= 613.9377216822213
ASTAR Expands  timisoara  f= 664.7083957336686
ASTAR Expands  arad  f= 592.4180538144219
ASTAR Expands  zerind  f= 553.0
ASTAR Expands  oradea  f= 520.0
Found goal
['drobeta', 'mehadia', 'lugoj', 'timisoara', 'arad', 'zerind', 'oradea']
```

It is seen here that the straight-line distance heuristic searches for the shortest path and abandons the path it was following upon reaching Arad because at that point the cumulative distance for Craiova was shorter than for Arad.

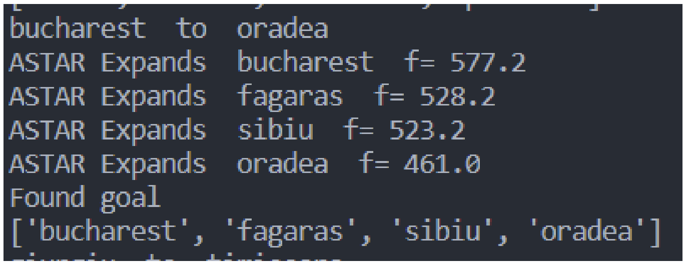
Manahan distance and sum heuristics have much longer distance as heuristic resulting in the outcome that is not optimal.

Straight line distance heuristic from Bucharest to Oradea opens 5 nodes and returns optimal path



```
bucharest  to  oradea
ASTAR Expands  bucharest  f= 408.7378622051057
ASTAR Expands  pitesti  f= 417.2662485944398
ASTAR Expands  rimnicu  f= 417.21870358160595
ASTAR Expands  sibiu  f= 430.6379376170944
ASTAR Expands  oradea  f= 429.0
Found goal
['bucharest', 'pitesti', 'rimnicu', 'sibiu', 'oradea']
```

Sum heuristic from Bucharest to Oradea opens 4 nodes and returns not an optimal path.



```
bucharest  to  oradea
ASTAR Expands  bucharest  f= 577.2
ASTAR Expands  fagaras  f= 528.2
ASTAR Expands  sibiu  f= 523.2
ASTAR Expands  oradea  f= 461.0
Found goal
['bucharest', 'fagaras', 'sibiu', 'oradea']
```

The average of the two heuristic returns 9 out of 10 optimal paths. The average of the two admissible heuristics is an admissible heuristic and is expected to produce optimal results. However, we have an average of admissible and inadmissible heuristics in this case, which is still performing better, providing a higher number of optimal results compared to the Manhattan distance heuristic alone.

For 10 city pairs selected Manhattan distance heuristic and Sum heuristic return the same number of optimal and not optimal paths, Average heuristic has a better performance and the Straight-line distance heuristic has the best results.