Ludwig-Maximilians-Universität München
Institut für Informatik
Lehrstuhl für Mobile und Verteilte Systeme
Dr. Juan Bernabé-Moreno

mobile and distributed systems group

Exercise Sheet 1
Intelligent Systems - WS 23/24

Exercise 1: Important Concepts

(Pkt.)

(a) Explain the following concepts: *Agent*, *Environment*, *Agent function*, *Rationality*, *Autonomy*, *Model based agent*, *Goal oriented agent*, *Benefit oriented agent*, *Reflex agent*

(b) Are reflexive actions - like pulling one's hand back from a hot stove top - rational, intelligent, or both?

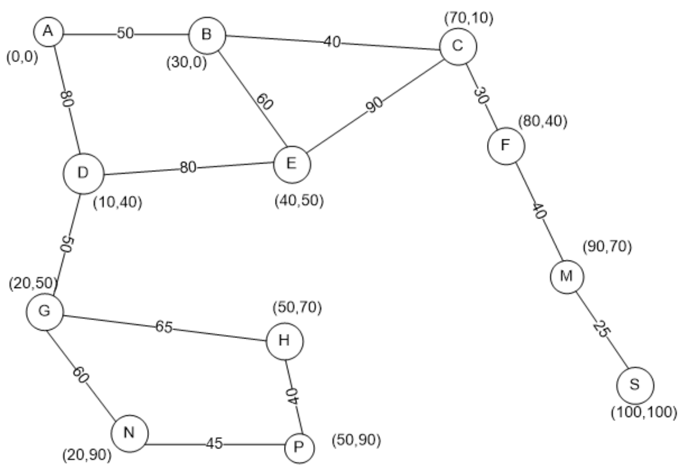
(c) Justify why it makes sense to formulate a problem after formulating the goal.

(d) Explain the following concepts: *State space*, *State*, *Search tree*, *Goal*, *Action*, *Successor function*, *Branching factor*

Exercise 2: Problem Formulation and Heuristics

(Pkt.)

Suppose two friends live in different cities on a map. In each turn, we can move both friends simultaneously to an adjacent city on the map. The time required to move from city i to neighbor j is equal to the road distance $d(i, j)$ between the cities. In each round, the friend who arrives first must wait until the other arrives before the next round can begin. We want the two friends to meet as soon as possible.



(a) Write a detailed formulation for this search problem

(b) Are there any fully connected (contiguous, not an “island”) maps for which there is no solution?

(c) Let $D(i, j)$ be the aerial distance between city i and j. Which of the following heuristic functions are admissible?

a. $D(i, j)$

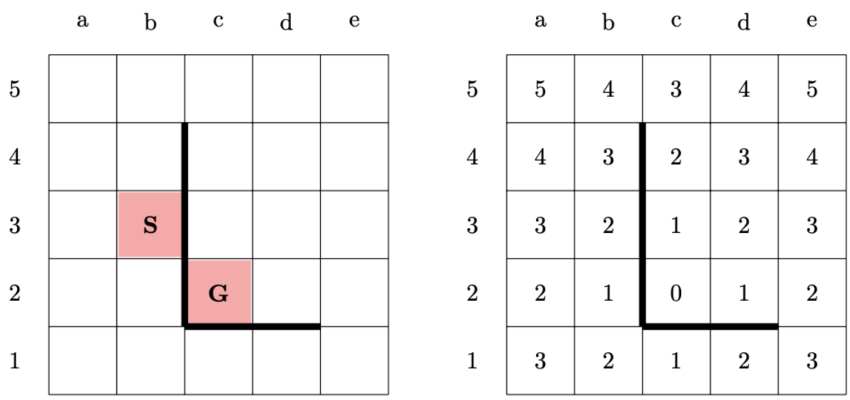
b. $2 * D(i, j)$

c. $D(i, j)/2$

Exercise 3: Application of A* (1)

(Pkt.)

A mobile robot wants to determine the shortest path from cell S (start) to cell G (goal). The robot can move around a horizontally or vertically connected cell at each step, as long as it is not separated from the current cell by a wall (thick black line). Each movement has a uniform cost of 1. The left figure shows the initial state, and the right figure the heuristic values.



(a) Perform an A* search to determine the shortest path from S to G. Enter the f-values of all generated nodes into the corresponding cells in the left figure. All other cells should remain empty.

(b) Is the heuristic admissible?

(c) How many nodes must A* expand when $h^*(n)$ is used as a heuristic, where $h^*(n)$ is the actual cost of an optimal path from cell n to goal G?

Exercise 4: Application of A* (2)

(Pkt.)

The 8-puzzle consists of 9 fields (3×3), on which 8 movable tiles are arranged; thus, one field remains free. Tiles that lie next to the free field can be moved into it. The goal is to arrange the tiles so that the following target state is reached:

1	2	3
4	5	6
7	8	

Run the A* algorithm to solve the 8-puzzle. Use the following start state:

1	2	3
	4	6
7	5	8

(a) Draw the search tree with the states visited by the algorithm as nodes. Also enter the corresponding f,g, and h values for each node. The heuristic to be used is the number of misplaced tiles ("misplaced tiles" heuristic). So for the initial state, the heuristic would estimate a cost of 3 (since tiles 4, 5, and 8 are misplaced). Moving a tile has an actual cost of 1. We assume that the algorithm remembers states that have already been visited and does not visit them again.

(b) What is the maximum cost that the heuristic from (a) estimates in the 8-puzzle?

(c) Another heuristic that can be used in the 8-puzzle is the *summed Manhattan-Distance* of all misplaced tiles to each tile's destination.

   a. Is there a state for which this heuristic estimates a lower value than the "Misplaced Tiles" heuristic?

   b. Give an example of a state where this heuristic estimates a higher cost than the "Misplaced Tiles" heuristic.

Exercise 5: Neighbor function for maximum clique and TSP

(Pkt.)

Familiarize yourself with the problems *Traveling Salesman* and *Maximum Clique*. How can a problem solution (a state) be represented for these problems? How can a neighboring state be constructed given another state?

Exercise 6: Tabu Search

(Pkt.)

Consider the following (undirected) graph G represented by an adjacency matrix:

0	0	1	0	1	1
0	0	0	1	1	1
1	0	0	0	1	1
0	1	0	0	1	1
1	1	1	1	0	1
1	1	1	1	1	0

a. Draw graph G

b. Name the objective function for Maximum Clique and the graph G

c. Maximize the objective function from (b) using Tabu Search. Use the starting state $[0, 0, 0, 0, 0, 0]$. Use a tabu list size of 3. Complete the following table until (including) iteration 8.

Iteration	$s_{current}$	value of $s_{current}$	tabu list	s_{best}	value of s_{best}
1	$[0, 0, 0, 0, 0, 0]$	0	$[[0, 0, 0, 0, 0, 0]]$	$[0, 0, 0, 0, 0, 0]$	0
2					

Exercise 7: Tabu Search - Coding

(Pkt.)

Given N players, each with individual playing strength s_i. Example: $s = [10, 5, 12, 39, 32, ...]$

Your task is to divide the N players into two teams of as equal strength as possible (the problem is also known as number partitioning). The team strength is the sum of all player strengths.

(a) Implement the *Tabu Search* algorithm in Python and use it to solve the above-presented task.

(b) Experimentally examine the influence of the player strength range on the difficulty of the optimization problem.