# Introduction to Intelligent Systems
# Project1: Rolling Die Mazes

Joseph Fuchs, Damien Cremilleux

October 8, 2014

## 1 Problem definition

The objective of this project is to solve rolling-die mazes of their optimal path. A die starts from a location on the maze, and can roll along its edges through a grid, until a goal location is found. The mazes can contain obstacles, and there are restrictions on which numbers are allowed to be face up on the die. The die will always start with the 1 facing up ('visible'), 2 facing up/north, and 3 to the right/east. Also, the number 6 should never be face up on the die at any position in the puzzle, and the number 1 must be on top of the die when the goal location is reached.

An example of maze is given below (S is the start location, G is the goal location and * are obstacles).

```
. G * . S .
. . . * . .
. * . . . *
. . . . . .
. . . * . .
```

This problem can be formulated as follow:

- States: The state of a rolling-die maze is determined only by both the die's position and orientation. So, a states description specifies the location (ie the coordinates) and the orientation (which number is facing up, north and east) of the dice on the maze.
- Initial state: The die is on the start location, with the 1 facing up, 2 facing north, and 3 facing east.
- Actions: At most, the possible actions on any state are to roll the die north, south, east or west. These actions result in the die moving in that direction by 1 board tile and in a corresponding change in the die's orientation. The valid subsets of these actions for any state depend on the orientation of the dice (the number 6 should never be turned face up),

and position on the board (the die should never move to the same tile as an obstacle or out of bounds).

- Transitions model: Given a state and action, this returns the resulting state.
- Goal test: The die is on the goal location, with the number 1 on the top.
- Path cost: Each action on a state costs 1 step, so the cost of a given path is measured by the number of steps in the path.

## 2 Heuristics

We use A* algorithm (discussed in the class) to solve this search problem. A* includes as a component of its evaluation function a heuristic function, $h(n)$. In the following sections, we describe the three heuristics chosen to solve this search problem.

### 2.1 Uniform-cost search

For the first heuristic, $h_1$, we chose a very simple solution: $h(n) = 0$. This means that the evaluation function is now $f(n) = g(n)$ where $g(n)$ is the cost to reach the node. With this heuristic, the A* search becomes a uniform-cost search algorithm. Uniform-cost search first visits the node with the shortest path costs from the root node.

This heuristic is admissible. It will never overestimate the cost to reach the goal because the heuristic is equal to 0, and path costs can not be negative.

Furthermore, this heuristic is consistent, which is a stricter requirement than admissibility. According to definition, a heuristic is consistent if, for every node $n$ and every successor $n$ of n generated by any action $a$, the estimated cost of reaching the goal from $n$ is no greater than the step cost of getting to $n$ plus the estimated cost of reaching the goal from $n$:

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

$0 \leq c(n, a, n)$

This is true because we are dealing with positive step costs.

### 2.2 Manhattan distance ignoring orientation

The second heuristic, $h_2$, is based on the city block distance, or Manhattan distance, which is the sum of the absolute differences of Cartesian coordinates between a state's die location and the goal location. This ignores obstacles, as well as orientation. To find this heuristic, we take the original problem with fewer restrictions: no obstacles and we do not take into account what the orientation of the die is, we just have to have it reach the goal square. We illustrate this on the following maze. The die is on the cell marked D. In this case, the heuristic will be $h(n = D) = \Delta x + \Delta y = 4 + 1 = 5$.

```
S . . . .
D . * * *
. * . . G
. . . . .
* . . . .
```

Because this is a relaxed problem of the first one, we know that the cost of an optimal solution to a relaxed problem is an admissible heuristic for the original problem. The cost estimate from any node to a goal will always be less than or equal to the true cost because the original, stricter problem must clearly have its die move at least $\Delta x$ in the horizontal direction and $\Delta y$ in the vertical direction to reach the goal state.

Additionally, this heuristic is also consistent, because it is the exact cost for the relaxed problem so it must obey the triangle inequality. Finally, this heuristic gives larger values than h1. This means that we can expect better results with this heuristics than the previous one.

## 2.3 Manhattan distance considering orientation

For the last heuristic, $h_3$, we also generated a heuristic from a relaxed problem. Similarly to heuristic two, we ignore obstacles. However, here we take the orientation of the die into consideration instead of ignoring it. Given the position and the orientation of the dice, it is possible to find a formula for the minimal number of actions to solve the maze from any location assuming there are no obstacles. For instance, if a die with 1 on top is on the same row or column as the goal position, the maze can be solve in $mahattan\_distance + 2$.

$Given:$

$\quad \Delta Y,\ the\ vertical\ distance\ from\ goal$

$\quad \Delta X,\ the\ horizontal\ distance\ from\ goal$

$h(n) =$

$\quad \Delta X + \Delta Y + 4 \quad if\ die's\ 1\ faces\ goal$

$\quad \Delta X + \Delta Y + 2 \quad if\ die's\ 1\ faces\ up\ and\ (\Delta X\ or\ \Delta Y\ =\ 0)$

$\quad \Delta X + \Delta Y + 4 \quad if\ die's\ 1\ faces\ up\ and\ not\ (\Delta X\ or\ \Delta Y\ =\ 0)$

$\quad \Delta X + \Delta Y + 2 \quad if\ die's\ 1\ faces\ away\ from\ goal\ and\ (\Delta X\ or\ \Delta Y\ =\ 0)$

$\quad \Delta X + \Delta Y \quad else\ if\ die's\ 1\ faces\ away\ and\ (\Delta X\ or\ \Delta Y\ =\ 1)$

$\quad \Delta X + \Delta Y + 4 \quad otherwise$

So we can generate a set of rules giving the position and orientation. We observe that each time the rule has the form $mahattan\_distance + cst$, where $cst$ is the number of move needed to put the 1 on top of the die and this number is directed linked to the orientation o the die. Due to *cts* the value given by $h_3$ will be at least as big as $h_2$ for all nodes. So this heuristic is likely to be more effective than the previous one. We illustrate this heuristic with the following maze, where the die D has 1 on its top. Now the heuristic will be $h_3(n = D) = 4 + 2 = 6$.

```
S . . . .
. . * * *
D * . . G
. . . . .
* . . . .
```

We again used a relaxed problem of the original one. For the same reasons like $h_2$, this heuristic is admissible and consistent.

## 3 Performance metrics

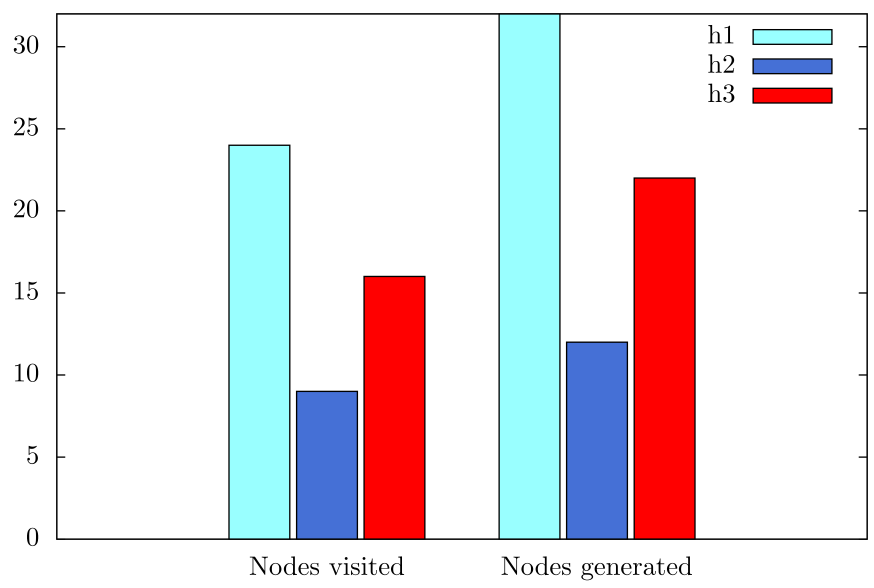
For each heuristics, we ran the five puzzles given in the wording. The results are in the following tables.

| | Puzzle 1 | | | Puzzle 2 | | | Puzzle 3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | $h_1$ | $h_2$ | $h_3$ | $h_1$ | $h_2$ | $h_3$ | $h_1$ | $h_2$ | $h_3$ |
| Number of nodes generated | 32 | 12 | 22 | 95 | 67 | 38 | 2 | 2 | 2 |
| Number of nodes visited | 24 | 9 | 16 | 84 | 50 | 34 | 3 | 3 | 3 |

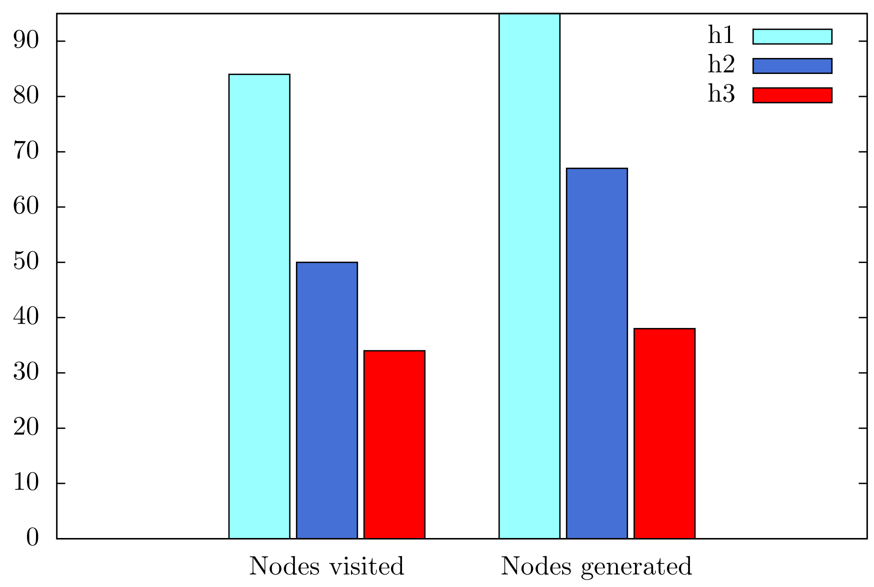
| | Puzzle 4 | | | Puzzle 5 | | |
|---|---|---|---|---|---|---|
| | $h_1$ | $h_2$ | $h_3$ | $h_1$ | $h_2$ | $h_3$ |
| Number of nodes generated | 161 | 101 | 81 | 1271 | 1080 | 163 |
| Number of nodes visited | 149 | 82 | 65 | 1260 | 737 | 98 |

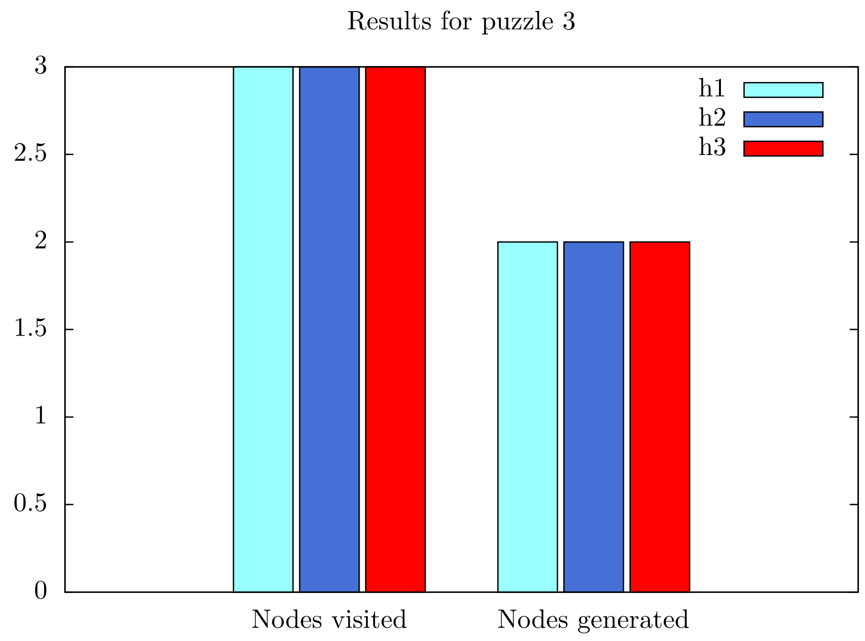
To show the difference, the following bar graphs give the number of nodes generated and visited according to the heuristic, for each puzzle.

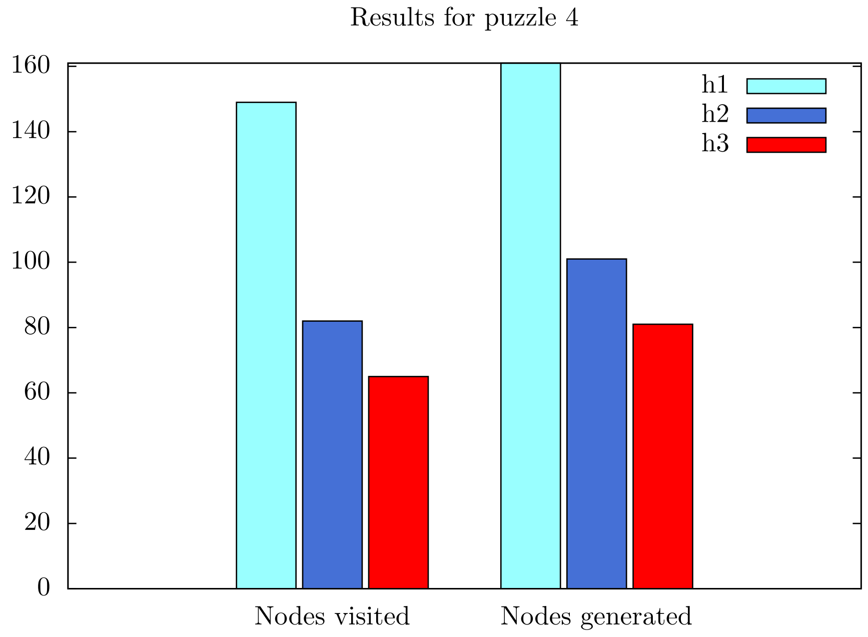
Results for puzzle 1



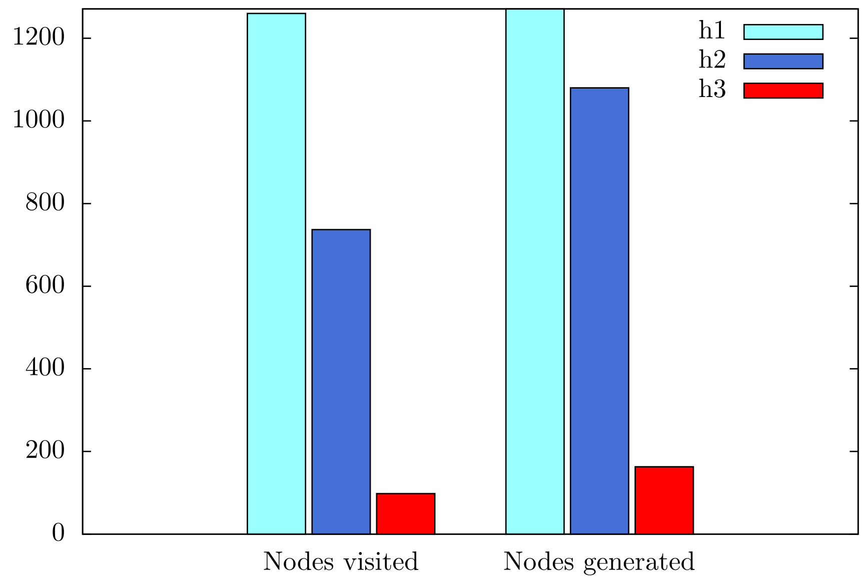
Results for puzzle 2




Results for puzzle 3
3
2.5
2
1.5
1
0.5
0
h1
h2
h3
Nodes visited
Nodes generated


Results for puzzle 4
160
140
120
100
80
60
40
20
0
h1
h2
h3
Nodes visited
Nodes generated

Results for puzzle 5



## 4 Discussion

The first observation we can make is that the first heuristic generates a huge amount of nodes, more than 1200 for the puzzle 5. The second heuristic is better, however the number of nodes generated is still huge (around the 1000 for the puzzle 5). According to our expectations, the third heuristic has the best results. The improvement are huge: with this heuristic, the number of nodes generated is only around 160 for the puzzle 5. We can explain the gap beetween the last two heuristics by the fact that the orientation of the grid has an important impact on the movements needed to be on the goal location with the 1 on the top.

For the more simple mazes, there is still a gap between the heuristics, but the difference is smaller. The third heuristic even generates more nodes than the second one for the first puzzle. The superiority of the third heuristic is really shown with complex mazes.

For the puzzle 3, we see that the number of nodes generated is less than the number of nodes visited, for all the heuristics. The explaination is that the initial node is visited, and not expanded. Only two nodes are expanded, and then the die is blocked.

As a conclusion, this project shows use the importance of the choice of heuristic for the A* algorithm. The quality of the heuristic has a huge effect on the performance of the algorithm.