Tuesday, 8 May 2018

EARIN Miniproject 2 - optimization

Group: P2.10 - Maciej Tutak, Łukasz Monecki

I. Puzzle rules

Kuromasu is a binary-determination logic puzzle. Kuromasu is played on a rectangular grid. Some of these cells have numbers in them. Each cell may be either black or white. The object is to determine what type each cell is.

The following rules determine which cells are which:

- Each number on the board represents the number of white cells that can be seen from that cell, including itself. A cell can be seen from another cell if they are in the same row or column, and there are no black cells between them in that row or column.
- Numbered cells may not be black.
- No two black cells may be horizontally or vertically adjacent.
- All the white cells must be connected horizontally or vertically[0].

II. Approach to problem solution

We consider a quintet <*S, P, S_0, T, c*>[1], where:

- *S* - search space
- *P* - set of state transition operators: P: S->S
- *S_0* - initial state
- *T* - set of terminal states
- *c* - cost function

These values fitted to our problem are as following:

S	Space of different board states
P	Generating a state with one more black cell than the state before
S_0	Board without any black cells
T	Every board state
T_f	A board state that is correct within the puzzle rules
c	Amount of steps taken to reach given state

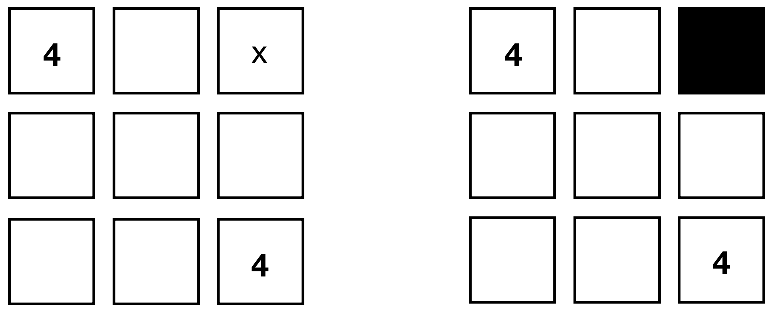
III. Heuristic function design

We consider a function: $f(x) = g(x) + h(x)$, where $f(x)$ is our objective function, $g(x)$ is a cost function and $h(x)$ is our heuristic[2].

An obvious candidate for the $g(x)$ is the number of steps taken to reach current state.

Since our algorithm will attempt to choose a path that minimizes *f(x)*, and at the beginning *g(x) = 0*, we need a function *h(x)* which value gets smaller, the closer we are to the solution. Therefore, we chose the amount of numbered cells which see correct amount of white cells for *h(x).* Unfortunately, this *h(x)* is not admissible, because sometimes it can overestimate the amount of steps needed to solve the puzzle.

A good example of this behaviour is such board configuration:



In this case, our *h(x) = 2,* despite the fact, that only one step is needed to complete this puzzle.

Perhaps, a better function could be a function of minimal amount of black cells needed to complete the puzzle, but calculating that value turned out to be complicated.

Since our heuristic function is not admissible it is also not monotone[4].

IV. Results

Results of the program are in a file called `out.txt` which was generated by running the script in a bash shell `run.py`: `$ python3 run.py > out.txt`

Examples are taken from a file `in.txt`.

V. References

- [0] - https://en.wikipedia.org/wiki/Kuromasu
- [1] - Problem solving by state-space searching, Paweł Zawistowski
- [2] - https://en.wikipedia.org/wiki/A*_search_algorithm
- [3] - https://en.wikipedia.org/wiki/Admissible_heuristic
- [4] - https://en.wikipedia.org/wiki/Consistent_heuristic