A* and the 8-Puzzle

Jackson Spell and **Ben Wiley**
{jaspell,bewiley}@davidson.edu
Davidson College
Davidson, NC 28035
U.S.A.

Abstract

An 8-puzzle is a tile puzzle which is solved by rearranging tiles via sliding to reach an ordered goal configuration. We use the A* local search algorithm to solve the 8-puzzle, comparing the behavior and efficiency of two relatively simple heuristic functions in order to evaluate the difference in puzzle generation cost and effective branching factor.

1 Introduction

The 8-puzzle is one of a number of relatively simple problems, known as toy problems, which arise from logic tasks and are used to illustrate algorithm behavior, but do not describe real-world problems. The sliding-block family of puzzles are difficult to solve, with large state spaces, but have easily established representations and heuristics and have been historically used to evaluate search algorithms (Ratner and Warmuth 1986).

The 8×8 puzzle has 181,400 reachable states (Russell and Norvig 2003). Previous analyses exploring larger puzzles have shown that solving the N×N puzzle is NP-hard and thus computationally intractable (Ratner and Warmuth 1986).

In this paper, we describe the heuristics used by the A* algorithm, detail our experimental approach, including our random board generation method, and then present the results of the comparison in heuristics.

2 Heuristics

We employ two commonly-used heuristics based on board state and tile positions relative to the goal (Russell and Norvig 2003). The purpose of this experiment is to evaluate the difference in efficiency of A* search between the two heuristics. The first heuristic tallies the number of tiles out of place on the board, not including the empty space, and the second sums the taxicab distance from each of the tiles to its target position in the goal configuration – the minimum number of moves needed to reach the target, ignoring the presence of other tiles. Both heuristics are admissible, as more moves are required to move all tiles to the goal configuration than the value of either heuristic.

3 Experiments

The objective in this experiment is to investigate the difference in search cost and effective branching factor between the two heuristic functions. We define the cost of the search to be the number of nodes generated – in this case, the number of boards created by moving tiles. The effective branching factor is calculated, for a solution of depth d and search cost of N, by

$$N + 1 = 1 + b^* + b^{*2} + ... + (b^*)^d$$

where b^* is the effective branching factor. This polynomial equation is solved using a numerical solver.

The heuristics are compared by averaging the search cost and effective branching factor for a given depth across 100 search operations, with the board randomly initialized each time. Because half of the possible board configurations are unsolvable (Johnson and Story 1879), we create the randomized board by starting in the goal configuration and making a number of random legal moves, thus ensuring we never enter an unsolvable state. The number of moves is itself a random number between 2 and 100 in order to fully populate the state space.

The possibility of cycles in the moves during the board randomization means that the actual solving depth is bounded on the upper side by the number of randomization moves. The actual depth is found by solving the board with one heuristic. The second heuristic runs only if 100 search operations of the depth have not already been completed; that is, the second heuristic runs only if more searches are needed at that depth in order to correctly average the algorithm's behavior.

4 Results

Present the results of your experiments. Simply presenting the data is insufficient! You need to analyze your results. What did you discover? What is interesting about your results? Were the results what you expected? Use appropriate visualizations. Prefer graphs and charts to tables as they are easier to read (though tables are often more compact, and can be a better choice if you're squeezed for space). **Always** include information that conveys the uncertainty in your measurements: mean statistics should be plotted with error bars, or reported in tables with a ± range. The 95%-confidence interval is a commonly reported statistic.

Embedding Pictures

See the source code (`results.tex`) for instructions on how to insert figures (like figure 1) or plots into your document.

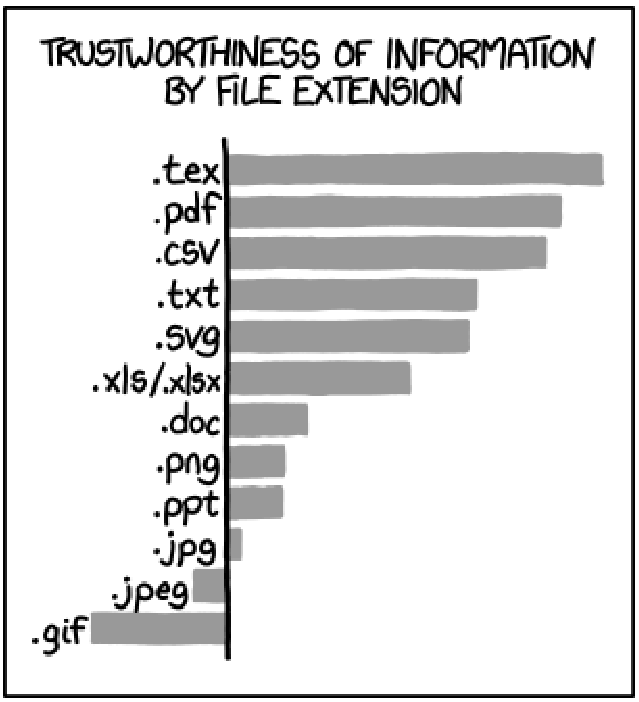

Figure 1: On the trustworthiness of LaTeX. Image courtesy of `xkcd`.

Creating Tables

Again, refer to `results.tex` to learn how to create simple tables (like table 2).

5 Conclusions

In this section, briefly summarize your paper — what problem did you start out to study, and what did you find? What is the key result / take-away message? It's also traditional to suggest one or two avenues for further work, but this is optional.

Column 1	Column 2	Column 3
1	3.1	2.7
42	−1	1729

Figure 2: An example table.

6 Acknowledgements

This section is optional. But if there are people you'd like to thank for their help with the project — a person who contributed some insight, friends who volunteered to help out with data collection, etc. — then this is the place to thank them. Keep it short!

References

Johnson, W. W., and Story, W. E. 1879. *Notes on the" 15" puzzle*.

Ratner, D., and Warmuth, M. K. 1986. Finding a shortest solution for the n× n extension of the 15-puzzle is intractable. In *AAAI*, 168–172.

Russell, S. J., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*. Pearson Education.