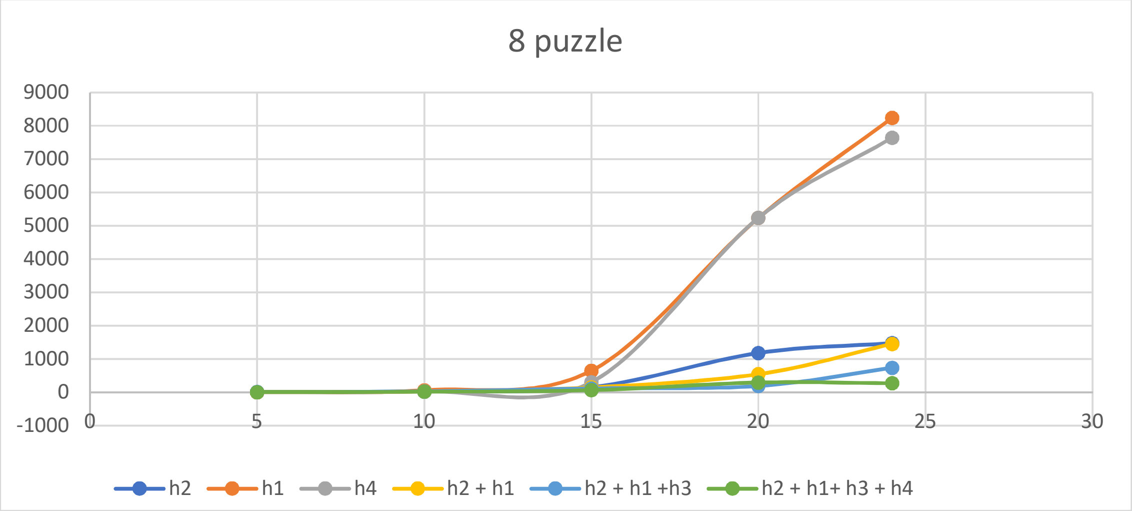
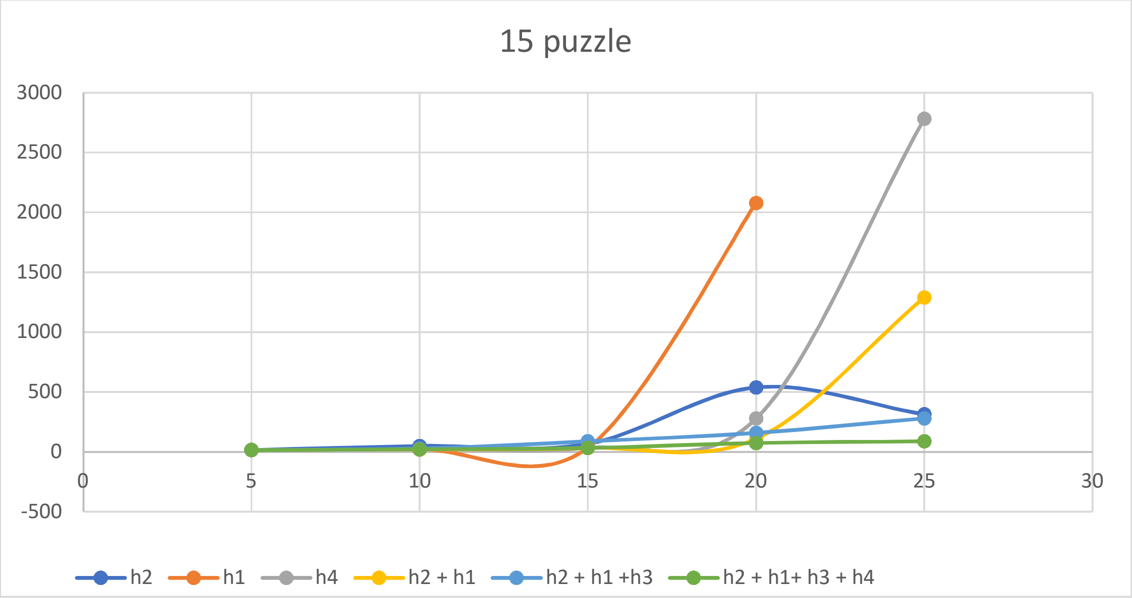
Observations and Findings:

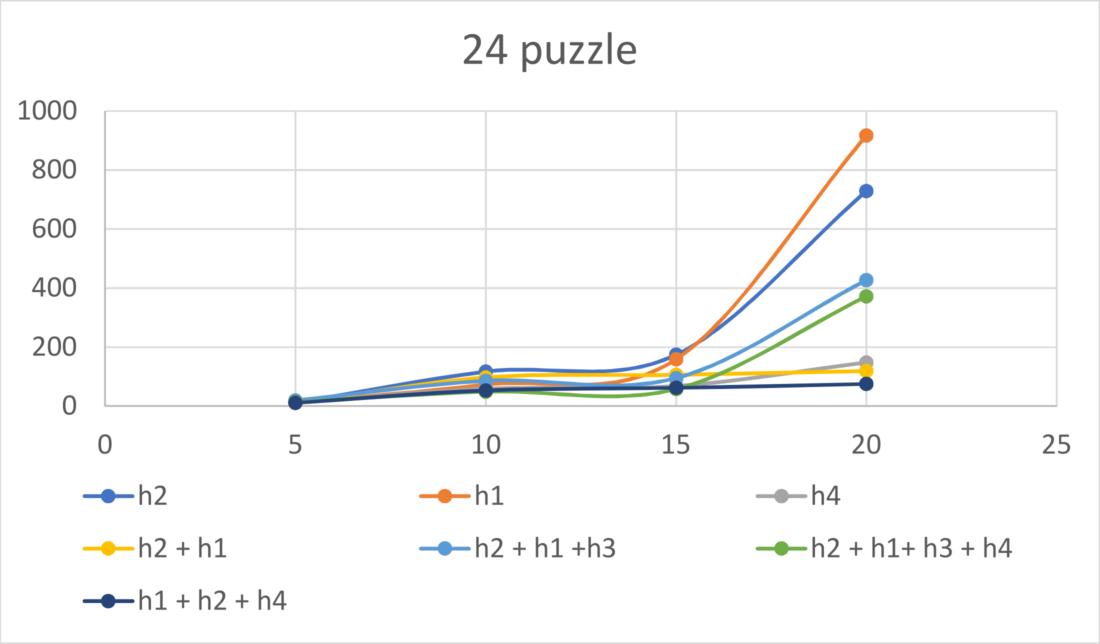


The above is the plot of complexity of a 3x3 puzzle to its computational cost. The X axis depicts the minimum number of steps required to solve the puzzle and Y axis is the number of unique nodes (states of puzzle) explored before reaching the final solution.

It can be seen that single heuristic functions such as h_1 (misplaced tiles) and h_4 (misplaced rows/column) ends up being at a very high cost for more complex or shuffled cases. However, the multi – heuristic function $h = h_1 + h_2 + h_3 + h_4$ solves the puzzle pretty much in constant time regardless of the complexity.



The above is a similar plot, this time with a 4x4 puzzle. There are a couple of things evident here. The performance of h_1 (misplaced tiles) at 15 step complexity is the best, however, at 20 steps it has an exceptionally high computational cost. Looking at the trend in general it is quite evident again that multi – heuristics are generally better at computing cost. Among the single heuristics h_2 (Manhattan distance) is probably the most efficient.



Finally, a 5x5 case. The heuristic function h_4 has done better than most multi – heuristic functions. Heuristic function h_3 (linear conflict) has given erroneous results effectively increasing the cost of every heuristic function it was a part of. Deleting it and creating a new heuristic function $h = h_1 + h_2 + h_4$ we find the results are much more appealing.

Conclusion:

- Admissibility of a heuristic function ensures the optimum result.
- The performance of a multi – heuristic function is generally better than a single heuristic function in terms of computational cost.
- A non-admissible multi – heuristic function might be efficient in terms of computational cost but the result might not be optimum. On the other hand, an admissible single heuristic would guarantee optimum solution however computational cost maybe high. Therefore, admissible multi – heuristic functions are desirable.
- 2 or more heuristic functions that performs poorly individually may give highly optimum results if incorporated in a multi – heuristic function with appropriate weights.
- For problems with low complexity performance of most heuristic functions are the same. The computational cost become more contrasting when the problem has a higher complexity.
- No heuristic function (single/multi) is universally better than other. The performance of a heuristic function may vary with different problems and situations.
- Usually more heuristics scales to a better computational cost, however, in certain situations some heuristics give erroneous cost estimates and thereby needs to be appropriately weighted.

References:

https://heuristicswiki.wikispaces.com/N+-+Puzzle
For the Heuristics’ definition

https://www.jaapsch.net/puzzles/javascript/fifteenj.htm
For input data

http://in.mathworks.com/help/matlab/
For various sub code/syntax

http://www.cs.cmu.edu/~venkatrn/papers/rss14.pdf
Project paper