Goals and Heuristics in Problem Solving

Logan Walls
University of Washington

In the context of problem-solving, a normative answer is not always practical for humans: we must often balance the costs of processing time, cognitive load, and accuracy by using heuristics. I conduct a between-subjects study which aims to explore whether or not being presented with different goals will affect human problem solvers' use of heuristics. I find that goal does indeed have an effect on heuristic use, and that a goal which explicitly emphasizes accuracy elicits solutions with greater resemblance to normative solutions.

Introduction

It is well established that the human problem solving process is greatly affected by the solver's representation of the problem and their understanding of the goal state. (Duncker & Lees, 1945) It has also been observed that subjects' can be representation of a problem can be influenced by the presentation of the problem. (Ahn & Graham, 1999) As humans must generally reason in time-constrained situations we have developed heuristics which allow us to sacrifice some our accuracy for increases in speed or decreases in cognitive load. (Gigerenzer & Selten, 2002) However the existence of formal logic and other normative tools is evidence that (under the right circumstances) humans are also capable of employing normative strategies under the correct circumstances. The goal of this study is to better understand how problem representation influences a problem solver's use of heuristics.

Design

For this study participants' performance was evaluated on the 8-Puzzle task, a variant of the famous 15-Puzzle. Participants are given a square grid containing numbered tiles and one empty space. The goal of the puzzle is to arrange the tiles in ascending order by sliding tiles into the empty space (fig. 1).

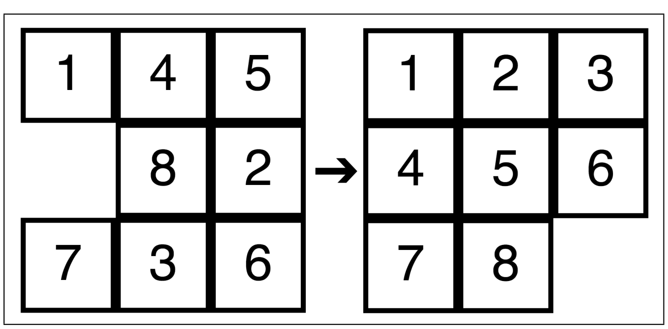

Figure 1. An example beginning state for the 8-Puzzle (left) and the goal state (right).

Procedure

Participants (n = 14, 6 male, 8 female, aged 19-25) was selected by convenience sampling. The study was conducted on a computer with most participants completing the experiment remotely using their own computers (however two subjects completed the study in person using the experimenter's laptop). Upon opening the website the participants were asked to fill out a brief demographic survey after which they were randomly assigned to one of two conditions: "speed" and "accuracy". Each condition received different instructions: in the speed condition participants were prompted to solve the 8-Puzzle as quickly as possible, while those in the accuracy condition were asked to solve the 8-Puzzle in as few moves as possible. After reading the instructions the participants solved a randomly shuffled 8-Puzzle (the sequence of moves chosen by the participant and the amount of time between moves were recorded during this process).

Analysis

To analyze participants' strategies, I compare each participant's sequence of moves to the sequences generated by six different computer algorithms. Five of these six algorithms are based on a depth-first search of the problem-space which prioritizes paths according to a distance function (a function which quantifies the difference between the current state is from the goal state). Each of these five algorithms attempts to simulate a different heuristic by using different distance functions.

The distance functions used by these algorithms are based on the Manhattan distance between each tile's current position and its position in the goal state:

$$m(S, S^*) = \sum_i \left|S^*_{ix} - S_{ix}\right| + \left|S^*_{iy} - S_{iy}\right| \tag{1}$$

Where S is a vector representing the current state, S_i is a coordinate pair ($row, column$) representing the i^{th} tile's position in the state. (S^* is a vector which follows the same structure, and contains the goal state).

This distance is supplemented with the linear conflict heuristic which increases the distance score when a tile's path to its goal position is obstructed by other tiles:

1. Assume two tiles $S_i = (x_i, y_i), S_j = (x_j, y_j)$ with goal positions $S_i^* = (x_i^*, y_i^*), S_j^* = (x_j^*, y_j^*)$.
2. A linear conflict occurs is any of the following are true:
   - $x_i = x_j = x_i^*$ and $y_i < y_j <= y_i^*$
   - $x_i = x_j = x_i^*$ and $y_i > y_j >= y_i^*$
   - $y_i = y_j = y_i^*$ and $x_i < x_j <= x_i^*$
   - $y_i = y_j = y_i^*$ and $x_i > x_j >= x_i^*$

Each algorithm was designed to emulate different strategies and heuristics which I hypothesized might be used by humans to solve the 8-Puzzle:

Algorithm	**Description**
Vanilla	Uses the Manhattan distance and penalizes both horizontal and vertical conflicts.
Rows First	Prioritizes placing each tile into the correct row by ignoring horizontal linear conflicts.
Columns First	Prioritizes placing each tile into the correct column by ignoring vertical linear conflicts.
Row Order	Prioritizes solving the rows in sequential order by weighing earlier rows more heavily.
Tile Order	Prioritizes solving the rows in sequential order by weighing earlier tiles more heavily.
Shortest Path	A breadth-first search of the problem-space, which yields the shortest possible solution.

(For details about each distance function see Appendix A)

To determine the similarity between a sequence of moves produced by the participant: p and a sequence of moves produced by an algorithm: a, I compute the Levenshtein distance(Levenshtein, 1966) between the two sequences of moves and normalize by the average of the sequences' lengths:

$$similarity(p, a) = 1 - \frac{lev_{p,a}(|p|, |a|)}{(|p|\,|a|)/2} \quad (2)$$

where *lev* refers to Levenshtein distance. The resulting score is valued between 0 and 1, where 0 implies the sequences are completely different and 1 implies they are exactly the same.

Results and Discussion

Grouping by condition I found that participants in the accuracy condition exhibited move-sequences significantly more similar to a shortest-path solution than any of the other algorithms, while move sequences from the speed condition more closely resembled the Vanilla (which uses a strategy akin to hill-climbing) and Tile Order algorithms (fig. 2). This supports the position that task goals influence problem-solvers' use of heuristics, as participants in the accuracy condition produce more normative solutions while those in the speed condition produce solutions which resemble the use of heuristics.

Additionally, for all five heuristic-based algorithms the sequences generated by those in the speed condition match the algorithm significantly more than those generated in the accuracy condition. Comparing the average amount of time taken per move across the conditions I found that participants in the accuracy condition took more time between moves than those in the speed condition. However, this difference becomes much smaller if the time taken before making the first move is excluded (fig. 3), suggesting that participants in the accuracy condition considered more possibilities before executing their strategy. This is consistent with the behavior of a breadth-first search, which considers every possibility from the current state before exploring other states.

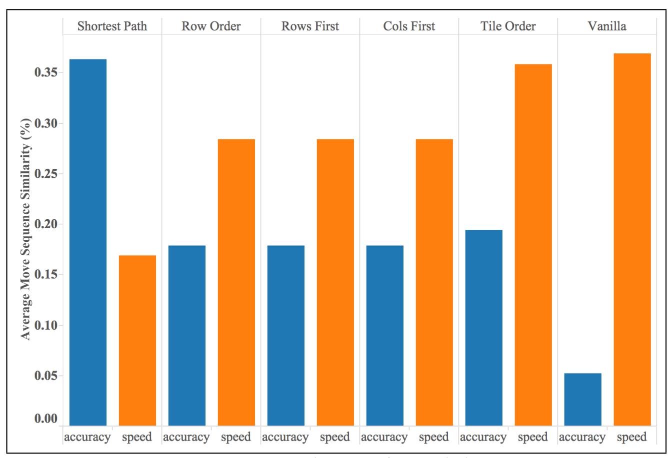

Figure 2. Average comparison of participants' move sequences from each condition to algorithm sequences.

Gender Effects

I also performed a participant-algorithm comparison grouping by gender and observed that on average females exhibit strategies which resemble the Tile Order algorithm while males exhibit strategies which are more similar to algorithms which solve the puzzle in chunks (Rows First, Columns First, etc.) (fig. 4). On average females also appear to spend less time between moves than males, and spend less time thinking before their first move (fig. 5). However both of these results are potentially biased by imbalances in the sample and condition assignment (see Limitations).

Limitations

The size and scope of the sample population is modest ($n = 14$, aged 19-25) and the sample was obtained by a potentially biased method (convenience sampling). Additionally because many of the participants completed the experiment remotely it was not possible to control the experimental environment (differences in participant's surroundings, potential distractions, different computer operating systems, etc.) and some participants admitted to multitasking while solving the puzzle. In regards to gender effects, the results were likely affected by uneven assignment to speed / accuracy conditions across genders. Finally, while significant differences in move sequences were observed between subjects (fig. 2) on average no group of strategies were more than 40% similar, suggesting that significant elements of the human participants' strategies may not be accounted for by the six algorithms used in this study.

Future Directions

In addition to replicating this experiment with a more robust sample, it may be fruitful for future work to explore more types of algorithms. As mentioned above, it appears that there are significant elements of human strategies for solving the 8-puzzle which are not present in the algorithms used by this study, I recommend that future studies examine human strategies further and attempt to identify these missing elements. A starting point for this may be examining humans' potential to look ahead one or more moves: the algorithms used in this study are only aware of the current state and do not simulate multiple moves ahead and I suspect this may be one source of discrepancy between participants' behavior and the algorithms.

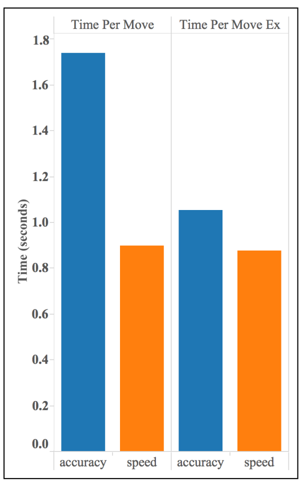

Figure 3. Average time taken per move by participants in each condition. Including (left) and excluding (right) lead-time before the first move.

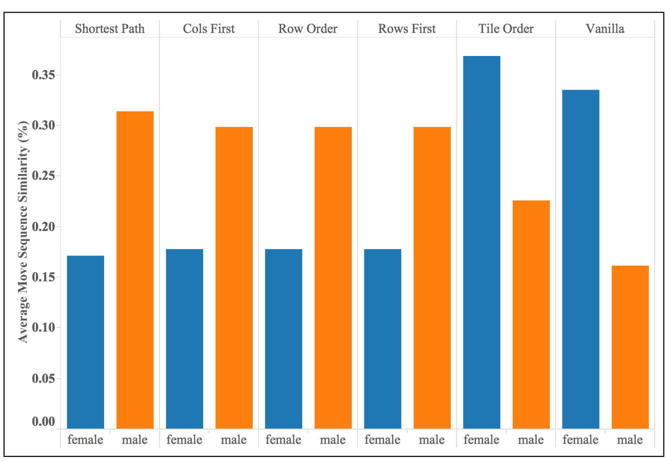

Figure 4. Average comparison of participants' move sequences from each gender to algorithm sequences.

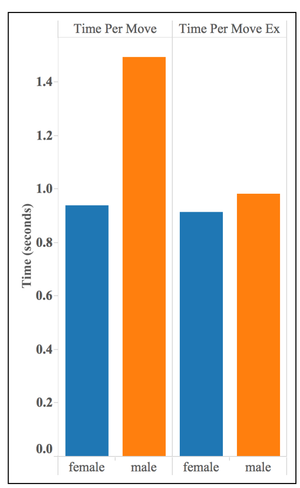

Figure 5. Average time taken per move by participants in each gender. Including (left) and excluding (right) lead-time before the first move.

References

Ahn, W.-k., & Graham, L. M. (1999). The impact of necessity and sufficiency in the wason four-card selection task. *Psychological Science*, *10*(3), 237–242.

Duncker, K., & Lees, L. S. (1945). On problem-solving. *Psychological monographs*, *58*(5), i.

Gigerenzer, G., & Selten, R. (2002). *Bounded rationality: The adaptive toolbox*. Mit Press.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady* (Vol. 10, pp. 707–710).

Appendix A

Table A1
Algorithm Distance Equations

Algorithm	**Distance Function**
Vanilla	$d(S,S^*) = m(S,S^*) + 2(C_h(S,S^*) + C_v(S,S^*))$
Rows First	$d(S,S^*) = m(S,S^*) + 2C_v(S,S^*)$
Columns First	$d(S,S^*) = m(S,S^*) + 2C_h(S,S^*)$
Row Order	$d(S,S^*) = \sum_{i} \left(3 - S_i^{*y}\right)\left(\left\|S_i^{*x} - S_i^x\right\| + \left\|S_i^{*y} - S_i^y\right\|\right) + 2(C_h(S,S^*) + C_v(S,S^*))$
Tile Order	$d(S,S^*) = \sum_{i=1} \left(9 - S_i^{*x} + 3(S_i^{*y} - 1)\right)\left(\left\|S_i^{*x} - S_i^x\right\| + \left\|S_i^{*y} - S_i^y\right\|\right) + 2\left(C_h(S,S^*) + C_v(S,S^*)\right)$
	Note: $C_h(S,S^*)$ is the number of conflicting tile-pairs in the horizontal direction, and $C_v(S,S^*)$ is the number or conflicting tile-pairs in the vertical direction.