What Determines Difficulty of Transport Puzzles?*

Petr Jarušek and **Radek Pelánek**
Masaryk University Brno, Czech Republic

Abstract

What determines difficulty of solving a problem? Although this question has been studied before, we found examples which show large differences in problem difficulty which are not explained by concepts identified in previous research. This differences are caused mainly by the structure of a problems' state spaces and cannot be easily captured by static metrics like size of the state space or the length of a solution. To address these unexplained differences, we propose a computational model of human problem solving behaviour. We provide evaluation of the model over large scale dataset (hundreds of hours of problem solving, more than 100 problem instances) for three transport puzzles (Sokoban, Rush hour, and Replacement puzzle).

Introduction

Human problem solving has been studied for a long time, starting by a seminal work by Simon and Newell (1972); for a recent overview see (Pizlo 2007). Simon and Newell (1972) and most of the subsequent research did use detailed data (e.g., think aloud protocols) about few hours of human problem solving. We use a complementary approach that allows us to address new questions: using modern technology we collect large scale data about hundreds of hours of human problem solving of more than 100 hundred problem instances.

In our analysis we focus on the issue of problem difficulty. What makes a problem difficult for humans? Why do two very similar problems differ in difficulty? These questions are important particularly in education – choosing problems of the right difficulty is very important to keep learners engaged. Easy problems are boring, difficult problems are deterring (Csikszentmihalyi 1975).

There are, of course, several factors that influence problem difficulty and many of them have been studied by researchers. At first, problem difficulty is influenced by the context of problem solving. The same problem may have different difficulty depending on the context in which it is presented. A typical example of this is the Einstellung effect (Luchins 1942) first studied for the water jug problem.

*This work is supported by GA ČR grant no. P202/10/0334.

At second, the overall problem difficulty depends on the difficulty of individual steps in the solution. This effect was demonstrated particularly with the use of izomorphic problems, i.e., problems which have the same underlying structure but different cover story. Most well-known are results for Tower of Hanoi (Kotovsky, Hayes, and Simon 1985) and Chinese ring puzzle (Kotovsky and Simon 1990). Different instances or representations of the same problem differ in requirements on working memory. The load on working memory directly influences the difficulty of overall problem solving.

At third, problem difficulty is influenced by the overall structure of the problem state space. Previous research has focused on straightforward measures like the size of the state space, the length of a solution, or the effectiveness of a hill climbing heuristic. These metrics were studied for river crossing problems (Greeno 1974), Fifteen puzzle (Pizlo and Li 2005), or Water jug puzzle (Atwood and Polson 1980; Carder, Handley, and Perfect 2008).

In this paper we report on experiments, in which we fix most of these previously identified factors. We randomize the order of examples, so that the order effect is minimized. We use 40 problem instances of the same size and with the same rules, i.e., the difficulty of individual steps is very similar. The problems vary with respect to state space size, solution length, and heuristic effectiveness, but we show that these factors do not explain the differences in problem difficulty.

We believe that these unexplained differences are caused by the structure of problems' state spaces. Figure 1 demonstrates on artificial examples how the structure can influence the difficulty. Both examples have the same number of states, edges, and the same distance from the start to the goal state. In the example on the left, it is easy to find the path to the goal – whatever path we choose we arrive at the goal. In the example on the right, it is much more difficult to succeed – we have to select the right sequence of moves and each wrong move makes a solution path much longer.

To capture these structural differences among problems, we propose a dynamic computational model which simulates human behaviour during state space search. The model is very abstract – it approximates human behaviour as a mix between randomness and optimality. The model does not provide explanation of "how people think", it just sim-

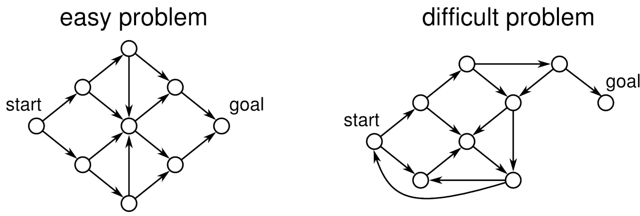

Figure 1: The structure of the state space can influence difficulty of problem solving.

ulates behaviour; i.e., it is a cognitive engineering rather than cognitive science model (Gray 2008). Cognitive science models provide a better explanation of experimental results, but they typically contain many problem specific rules and many parameters, which makes them prone to data overfitting (Roberts and Pashler 2000) and unsuitable for generalization. Our model operates only with the problem state space and thus it can be easily applied to other state space traversal problems.

Using the computational model we specify a metric for rating difficulty of problems and we evaluate this metric over the collected data. We compare this metric with other possible difficulty rating metrics, particularly with the metric ‘the length of the shortest path to goal’. Results differ for the three studied puzzles. Based on these results we propose new hypotheses about human problem solving.

Experiments with Human Problem Solving

In this section we describe the methodology used for data collection on human problem solving. We also discuss the problems that were used and we provide a brief summary of the collected data.

Data Collection

We have performed data collection using the internet, more specifically using own web portal through which participant solve problems. All actions of problem solvers (including their timing) are saved and stored into a database on the server. Before solving experimental problems, participants solved few training problems to get acquainted with the rules. The order of experimental problems was randomly shuffled for each participant.

Participants were mainly university students and were not paid for solving. Since the whole experiment was ran over the internet we did not have direct control over participants. As a motivation to perform well there was a public results list – this is for most people sufficient motivation to perform well, and at the same time it is sufficiently weak so that there is not a tendency to cheat. Even if participants cheated, we would be able to recognize it, as the solving time and state space navigation would be significantly different from other data. For a more detailed analysis of the data see (Jarušek and Pelánek 2010).

This internet based approach has certainly some disadvantages over the standard ‘laboratory’ approach to experiments with human problem solving. Nevertheless, we believe that the advantages significantly outweigh these disadvantages. We have been able to collect extensive data about human problem solving activity: several hundreds people solved several thousands puzzles and spent more than 400 hours with problem solving. This is much more than would be feasible with the classical laboratory approach. Moreover, the experimental setting is cheap and the data collection fast.

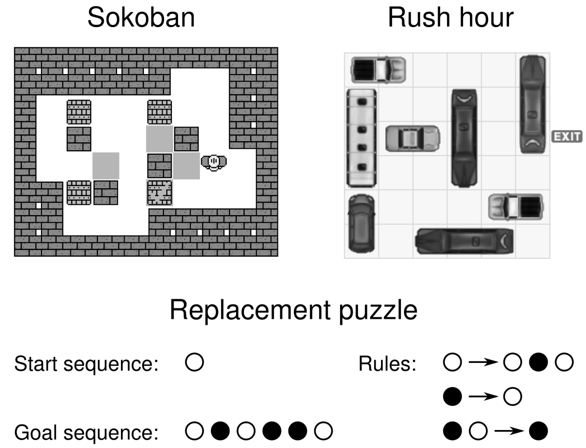

Figure 2: Examples of studied puzzles

Studied Problems

We focus on well-structured problems, i.e., problems with clear objective and clear set of rules. A typical example of well-structured problems are logic puzzles. Puzzles are well suited for research, since they contain all important information in the statement of the problem (and hence do not depend on the knowledge of the solver), are amenable to automated analysis, and are also attractive for humans. For these reasons puzzles have been used for a long time in both artificial intelligence (Schaeffer and Van den Herik 2002) and cognitive psychology (Simon and Newell 1972; Wilson and Keil 1999).

All studied problems are single player transport[1] puzzles. Transport puzzles can be expressed directly using the state space terminology (Simon and Newell 1972) – states are configurations of the puzzle, transitions are given by allowed operations, the goal of the puzzle is to find a path from the start to the goal state. We study three puzzles (see Figure 2 for illustration): Sokoban, Rush hour, and Replacement puzzle. In the following we briefly describe rules of the puzzles, used instances, and we mention characteristics of their state spaces and summary of results of our experiments. Summary information about puzzles and collected data is provided in Table 1.

Sokoban Sokoban is a well-known puzzle created by Hiroyuki Imabayashi. There is a simple maze with several boxes and one man. The goal of the puzzle is to get the

[1] The “transport” notation does not mean that there is necessary some physical movement involved in solving the puzzle, but rather that the solution is a sequence of moves.

Table 1: Summary information about collected data. “Total time” is the total time spent by solvers to solve provided puzzles (only successful attempts are included).

problem	state space		instances	total time	median time to solve easiest	median	hardest
Sokoban	large	directed	35	356 hours	28 sec	10 min	56 min
Rush hour	large	undirected	45	46 hours	15 sec	2 min	11 min
Replacement puzzle	small	directed	40	55 hours	34 sec	2 min	5 min

boxes into the target squares. The only allowed operation is a push by a man; he can push only one box at a time. For experiments we used 35 instances, all of them with 4 boxes and similar size of the maze. Most of the instances were selected from standard level collections. State spaces are directed (moves are irreversible), their size ranges between 3000 to 36 000 states. Median time to solve a puzzle is 30 seconds for the easiest instance and nearly 1 hour for the hardest instance.

Rush Hour Rush hour is a well-known puzzle created by Nob Yoshigahara. In a grid there are several cars. Each car can move either in vertical or horizontal direction, cars cannot be rotated. Each square of the grid can be occupied by at most one car. The goal of the puzzle is to get a special red car out of the grid. Experiments were performed with 45 instances, all of them using 6×6 grid and cars of the size 1×2 or 1×3. Most of the instances were taken from the standard Rush hour set. State spaces are undirected (all moves are reversible), their size ranges from 600 to 80 000 states. Median time to solve a puzzle is 15 seconds for the easiest instance and 11 minutes for the hardest instance.

Replacement Puzzle Replacement puzzle is a lesser-known puzzle created by Erich Friedman. In this case we are manipulating a sequence of symbols. Given a starting sequence of symbols, the aim is to derive a goal sequence by using provided replacement rules. Replacement rules are applied one at a time; replacement can be applied on any consecutive sequence of symbols. At any time there may not be more than 6 symbols. Original formulation by Erich Friedman requires that the puzzle is solved in a fixed number of steps (to ensure a single solution), we allow arbitrary number of steps. The experiments were done with 40 instances; each of them used two types of symbols and three rules. State spaces are directed (moves are irreversible) and their size ranges between 10 and 120 states, i.e., in this case state spaces are much smaller than for Sokoban and Rush hour. Nevertheless, the puzzle is still nontrivial, median time to solve a puzzle is 30 seconds for the easiest instance and 5 minutes for the hardest instance.

Computational Model of Human Behaviour

Results of our experiment show that there are very large differences in problem difficulty even between very similar problems. One of the causes of these differences is the structure of underlying problem state space, as illustrated in Figure 1. But how do we measure the structure of the state space?

To address the impact of the state space structure on problem solving, we propose to use a dynamic computational model. We do not try to model actual human cognitive processes while solving the problem, but only to capture human behaviour during a state space traversal. Our model is very abstract and is based only on information about underlying problem state space, i.e., the model is not specific for a single problem.

Basic Model

Our web portal stores detailed data on human behaviour during problem solving. Analysis of the collected data showed that at the beginning of problem solving humans explore the state space rather randomly, whereas later, as they get closer to the solution, they move more straightforwardly to the goal.

Our computational model is based on this observation. The model starts at the initial state and then repeatedly selects a successor state. This selection is very simple – it is a combination of two tendencies: “random walk” (selection of a random successor) and “optimal walk” (selection of a successor which is closer to a goal state). Human decisions are usually neither completely random, nor completely optimal. Nevertheless, the model assumes that a weighted combination of these two tendencies can provide a reasonable fit of human behaviour.

The general principle of our model is the following. In each step the model considers all successors s' of the current state s. Each successor s' is assigned a value $score(s')$, the sum of all $score$ values is denoted $SumScore$. The model moves to a successor s' with a probability $P(s') = score(s')/SumScore$.

This general model is further specified by a selection of a $score$ function. The basic version of the model uses a simple scoring function based on distance[2] $d(s)$ of a state s from the nearest goal state. The function is defined as follows (B is a single parameter of the model – ‘an optimality bonus’):

$$score(s') = \begin{cases} d(s) & d(s') \geq d(s) \\ d(s) + B & d(s') < d(s) \end{cases}$$

Successors that lead towards a solution get an ‘optimality bonus’, i.e., they have higher chance of being selected. The use of distance from a goal has the consequence that the relative advantage of the bonus increases as the model

[2]Note that this is not a heuristic estimate, but just a plain graph distance in a state space. All our problems have state space smaller than 100 000 states, so we can afford to compute exact distance for each state.

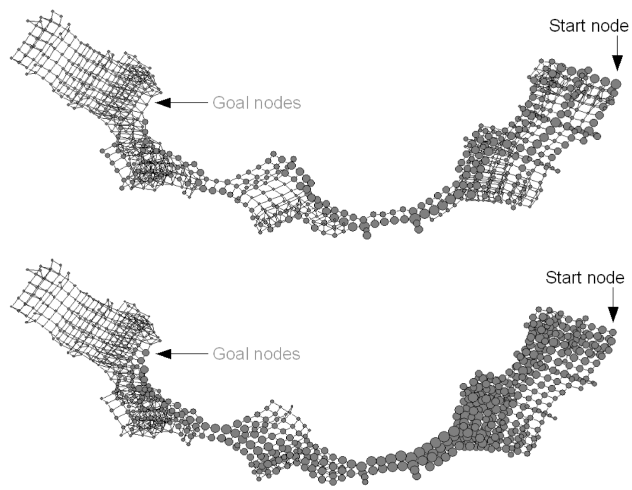

Figure 3: Comparison of traversal of the state space of one of the Rush hour problems by humans (upper image) and by computational model (lower image). The size of each state is proportional to the time spent in a given state.

gets closer to the goal, i.e., the model behaves less randomly when it is closer to the goal (as do humans). Figure 3 gives an illustration of behaviour of humans and computational model for one of the Rush hour problems.

If $B = 0$ then the model behaves as a pure random walk. As B increases, the behaviour of the model converges to an optimal solution. Hence by tuning the parameter B the model captures continuous spectrum of behaviour between randomness and optimality. Improvement in problem solving skills (e.g., by training or by studying worked examples) thus corresponds to increase in the value of B.

Extensions

When the state space is directed (as is the case for Sokoban and Replacement puzzle), it is not possible to reach a goal state for some states – we call these states 'dead'. Once the model reaches a dead state, it will forever cycle in dead states. Since this does not correspond to human behaviour, we have to extend the model for directed state spaces. We consider two different extensions:

1. dead states are never visited, i.e., $score(s') = 0$ if s' is dead;
2. the model resets back to initial state when it reaches a state without any successor or when it revisits a same dead state for second time.

We use the first extension for a Sokoban model, and second extension for the Replacement puzzle. This choice is based on the collected data about human problem solving. Humans are good at avoiding dead states in the Sokoban puzzle, whereas in the Replacement puzzle humans do visit dead states.

A natural extension of the model is the employment of hill climbing heuristic (favouring moves which improve the perceived distance between the current state and the goal). Previous research (Greeno 1974; Pizlo and Li 2005; Atwood and Polson 1980; Carder, Handley, and Perfect 2008) suggests that humans often use such heuristics during problem solving. The model can be simply extended by specifying a heuristic function[3] $h(s)$ and adding to $score(s')$ a 'heuristic bonus' whenever $h(s') < h(s)$.

Analogically, it is possible to extend the basic model by other heuristics, e.g., loop avoidance heuristic (the model remembers states that were already visited and the scoring function have higher values for unvisited states) or penalization of long back edges (humans can often recognize not just moves which lead to dead states, but also moves which lead "backwards").

Evaluation and Discussion

In this section we evaluate the model over the collected data. We also discuss the interpretation of these results and describe some hypotheses that the data suggest.

Difficulty Rating Metrics

Does the computational model provide an explanation of differences in problem difficulty? To answer this question we formalize a metric based on the computational model and compare it with other possible metrics. The metric based on the computational model works as follows: for a given problem we run the model repeatedly[4] over the state space and compute the mean number of steps necessary to reach the goal state.

For comparison we used several other metrics, e.g., parameters of a state space (particularly size), length of the shortest path to the goal state, and metrics based on simple heuristics like the number of counterintuitive moves (Carder, Handley, and Perfect 2008) that are necessary to reach a goal. From these other metrics we report here only the length of the shortest path, because metrics based on state space parameters do not provide a statistically significant correlation with problem difficulty, and metrics based on problem specific heuristics work similarly as metrics based on the shortest path[5] and as they are problem dependent we do not discuss them in detail.

Thus we focus on comparison of the shortest path metric and the computational model metric with human solving times. Figure 4 provides scatter plot for both metrics and Table 2 provides summary of correlation coefficients. Except for the standard Pearson's correlation coefficient, we also report Spearman's correlation coefficient, which gives the correlation with respect to ordering of values – for practical application of difficulty metrics the ordering is often more

[3]An estimate of the distance of s from goal state, e.g., for Sokoban a natural heuristic is the Manhattan distance of boxes from goal positions.

[4]The reported results are based on 1000 repetitions. We checked that the results are stable and are not changed by further increase in number of repetitions.

[5]The only notable exception is for Sokoban puzzle where we were able to get successful problem specific metric based on 'chunks' along the shortest path (Jarušek and Pelánek 2010).

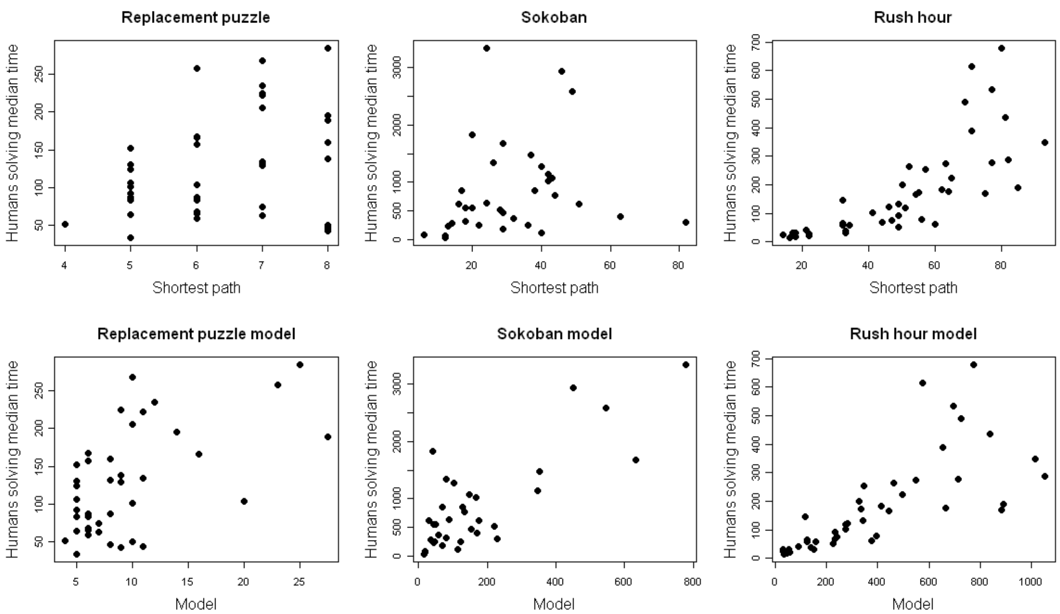

Figure 4: Scatter plots for shortest paths and model metrics. In 'Sokoban model' case there is one outlier which is outside the shown plot.

important than absolute values. Moreover Pearson's coefficient is in some cases rather unstable. This occurs particularly for the Sokoban puzzle, where removal of one outlier instance increases the Pearson's correlation coefficient from 0.39 to 0.80. Results for Spearman's coefficient are stable.

Sensitivity Analysis and Model Extensions

The metric based on computational model is dependent on the parameter B (optimality bonus). We have done sensitivity analysis of the model behaviour with respect to this parameter and the results show a surprising result. In all cases the best behaviour of the model is for values of the optimality parameter B around 25. Note that the three studied problems are quite different – their state spaces are different combinations of large/small and directed/undirected types (see Table 1). Is this an accidental result caused by our selection of problems? Or does the model and the specific value 25 of the optimality parameter tell us something general about human problem solving?

We have also performed evaluation of the model extension with hill climbing heuristics. The hill climbing heuristic is problem specific, so it is not possible to do straightforward comparison across different problems. Thus we focused only on Sokoban puzzle[6]. The correlation with human results improved to 0.56 (Pearson's coefficient), respectively to 0.73 (Spearman's coefficient).

[6]For Sokoban it is straightforward to specify a natural hill climbing heuristic based on Manhattan distance of boxes from goal positions. For the other two puzzles the choice of heuristic is not so clear.

Differences among Problems

Table 2 shows that there are quite large differences among the three studied problems. For Rush hour the shortest path metric provides a good explanation of problem difficulty, in this case the computational model metric does not bring any improvement. However, for the Sokoban puzzle and particularly for the Replacement puzzle, the shortest path metric provides poor explanation and in these cases the computational metric does bring an improvement.

These results thus open new interesting question: Why does the shortest path metric sometimes provide sufficient explanation of problem difficulty and sometimes it does not? To answer this question it is necessary to study more than just three different problems. Thus so far we can provide only several hypotheses.

At first, solving times for individual instances differ more significantly for Rush hour than for Sokoban and Replacement puzzle This difference is probably caused by selection of particular instances and by smaller variation in times for Rush hour puzzle. Thus to certain extent the differences can be artifacts of our experiments (it is easier for metrics to distinguish more significantly different problems), but we believe that this potential artifact can account only for small part of differences.

At second, problems differ in their "local difficulty". It is much harder to imagine successor states for the Replacement puzzle than for Rush hour puzzle. Thus solvers can do more analysis and planning for Rush hour and thus the structural differences among problems may not be that much important.

At third, the state space of Rush hour is undirected (all moves are reversible) whereas state spaces for Sokoban and

Table 2: Difficulty rating of puzzle instances for our three problems using the shortest path metric and the computational model metric.

		correlation coefficient	
problem	metric	Pearson's	Spearman's
Rush hour	shortest path	0.77	0.90
	model $B = 25$	0.75	0.90
Sokoban	shortest path	0.19	0.41
	model $B = 25$	0.39	0.61
Replacement puzzle	shortest path	0.28	0.21
	model $B = 25$	0.57	0.49

Replacement puzzle are directed[7]. We believe that the issue of directionality may be quite important in the study of problem solving. So far this issue was not adequately addressed in previous research, as most research did focused on undirected problems (Greeno 1974; Kotovsky, Hayes, and Simon 1985; Pizlo and Li 2005).

Conclusions

Using an internet, we collected large scale data about human problem solving of transport puzzles (Sokoban, Rush hour, Replacement puzzle). The data show that there are large differences among difficulty of individual problem instances. We argue that these differences are caused by global structure of problem state space and that they are not explained by previous research. In order to explain these differences, we develop a computational model of human behaviour during state space traversal. The model is a simple combination of random and optimal behaviour. It has just one parameter and the optimal value of this parameter is nearly the same for all three studied problems. Thus we believe, that the model should be easily applicable to other problems.

We evaluated the model over collected data and compared it with other metrics for difficulty rating. The results differ for the three studied puzzles. In the case of the Rush hour puzzle, it is easy to predict difficulty even with the shortest path metric. For the Replacement puzzle, simple metrics do not work and the computational model does bring a significant improvement. The results open several new questions about problem solving, e.g., the role of directionality, or interaction between difficulty of local steps and global structure.

The model can be also used for modeling improvement in problem solving skills by training – improvement corresponds to increase in the value of the model parameter B. Preliminary evaluation using the data from Tower of Hanoi experiment (Gunzelmann and Anderson 2003) showed that the model can provide good fit to previously published data.

Acknowledgement We thank Ondřej Bouda and Tomáš Křepský for their assistance with data collection.

[7] Although we did provide solvers a 'back' button for the Sokoban puzzle.

References

Atwood, M., and Polson, P. 1980. Further explorations with a process model for water jug problems. *Memory & Cognition* 8(2):182–192.

Carder, H.; Handley, S.; and Perfect, T. 2008. Counterintuitive and alternative moves choice in the Water Jug task. *Brain and Cognition* 66(1):11–20.

Csikszentmihalyi, M. 1975. *Beyond boredom and anxiety*. Jossey-Bass.

Gray, W. D. 2008. *The Cambridge Handbook of Computational Psychology*. Cambridge University Press. chapter Cognitive Modeling for Cognitive Engineering, 565–588.

Greeno, J. 1974. Hobbits and orcs: Acquisition of a sequential concept. *Cognitive Psychology* 6(2):270–292.

Gunzelmann, G., and Anderson, J. 2003. Problem solving: Increased planning with practice. *Cognitive systems research* 4(1):57–76.

Jarušek, P., and Pelánek, R. 2010. Human problem solving: Sokoban case study. Technical Report FIMU-RS-2010-01, Masaryk University Brno.

Kotovsky, K., and Simon, H. 1990. What Makes Some Problems Really Hard: Explorations in the Problem Space of Difficulty. *Cognitive Psychology* 22(2):143–83.

Kotovsky, K.; Hayes, J.; and Simon, H. 1985. Why are some problems hard? Evidence from tower of Hanoi. *Cognitive psychology* 17(2):248–294.

Luchins, A. 1942. Mechanization in problem solving: The effect of Einstellung. *Psychological Monographs* 54(6):95.

Pizlo, Z., and Li, Z. 2005. Solving combinatorial problems: The 15-puzzle. *Memory and Cognition* 33(6):1069.

Pizlo, Z. 2007. Human Problem Solving in 2006. *The Journal of Problem Solving* 1(2):3.

Roberts, S., and Pashler, H. 2000. How persuasive is a good fit? A comment on theory testing. *Psychological Review* 107(2):358–367.

Schaeffer, J., and Van den Herik, H. 2002. Games, computers, and artificial intelligence. *Artificial Intelligence* 134(1-2):1–8.

Simon, H., and Newell, A. 1972. *Human problem solving*. Prentice Hall.

Wilson, R., and Keil, F. 1999. *The MIT encyclopedia of the cognitive sciences*. MIT Press.