A Structural Complexity Measure for Predicting Human Planning Performance

Marco Ragni, Felix Steffenhagen, Thomas Fangmeier
{ragni, steffenhagen, fangmeier}@cognition.uni-freiburg.de
Center for Cognitive Science, Friedrichstraße 50
University of Freiburg, Germany

Abstract

Humans have an impressive ability to solve even computationally complex problems. Limited cognitive processing capabilities, however, impede an exhaustive search of the problem space. Thus, planning problems of the same size may require a different cognitive effort. Formal complexity aspects are inherent to a problem and set computational limits that a solver must deal with. For a measure of cognitive complexity, operational aspects of human cognition must be taken into account. We present a structural complexity measure for predicting human planning performance. This measure is based on the number and connectedness of subgoals necessary to solve a problem. This measure is evaluated on the PSPACE-complete puzzle game Rush Hour and is able to capture empirically measured difficulty for this game.

Keywords: Planning, Cognitive Complexity.

Introduction

Planned and rational behavior are daily aspect in everyday life. Planning can be defined as, the anticipation of action steps or "a procedure for achieving a particular goal or desired outcome" (Morris & Ward, 2005, p. 1). In computer science, one distinguishes between optimal and satisfiable planning. The goal of optimal planning is to find a shortest possible solutions for a problem, whereas the goal of satisfiable planning is to find a solution at all.

In AI and Cognitive Science finding a solution is often represented as a search of the problem space (Russell & Norvig, 2003). The problem space is defined by the operators and problem states. Due to limited cognitive processing resources, humans are not able to search the problem space exhaustively, i.e., they do not apply any operator on any state. Humans are, nonetheless, able to solve computationally complex problems by chunking information, in order to reduce the problem representation, (Ellis & Siegler, 1994; Kotovsky, Hayes, & Simon, 1985) and by applying heuristic search strategies (Miller, Galanter, & Pribram, 1960).

Planning problems have various characteristics. Problems can be non-transparent, have multiple goals, can be solvable, well-defined, dynamic, or decomposable. Another important issue is the domain of the problem. A first measure of the difficulty of a planning problem is the minimum number of steps necessary to solve the problem. In AI the different degrees of difficulty for problems are mostly classified according to the number of computing operations or the amount of memory required to solve a problem. For an overview of the complexity of planning tasks please refer to Helmert (2008). These measures are asymptotically with respect to worst case boundaries for increasing problem sizes (Papadimitriou, 1994). However, computational complexity measures do not integrate local problem structures. This is important for a more detailed measurement of problem difficulty, because problems with shorter solution length can be more difficult to solve for humans.

If too many operations are necessary, most humans seem to become overstrained, i.e., they make significantly more errors, need more time, and even start to guess. The difficulty for humans in solving planning problems can differ with regards to solvability, optimality, and response times. This implies, that there must be further problem-inherent planning differences which influence the performance of humans.

This aspect is important for explaining varying cognitive effort as it occurs in human problem solving. A cognitive complexity measure (a formal measure which is able to capture the human planning complexity) must not only integrate formal aspects of complexity, but also particularities of the human reasoning and planning process, e.g., the abundant use of heuristics or preferred operations.

We will define our cognitive complexity measure and evaluate it on (spatial) permutation problems like Rush Hour[1]. This planning problem developed by Nob Yoshigahara is a game with a visual-spatial presentation which is well-defined, solvable, decomposable, not dynamic and has only one goal. Given these settings, the number of operations can be controlled systematically and measured precisely. These planning problems have an initial state, an explicit goal state (e.g. where a certain relation must hold), and a number of underlying operations. Compared to Tower of London, Rush Hour has advantages, which guide the decision to use the latter: the problem size is easier to adjust, it has two dimensional features, the number of interacting objects is higher, which increases the difficulties for human reasoning (branching, counterintuitive moves), and Rush Hour is PSPACE-complete (Flake & Baum, 2002) and sufficiently complex for our purposes. It is also possible to generate highly challenging problems. Thus, it is important to find parameters, which describe more precisely difficulties humans encounter in planning tasks as was possible in classical theoretical computer science

In the following we first analyze (formal) requirements of a cognitive complexity measure to capture the average human planning process and introduce a first notion of a structural complexity measure. This is exemplified on the PSPACE-complete puzzle game Rush Hour. This structural complexity measure, although defined formally, is able to capture empirically measured difficulty for this game. Identified solution strategies and examples conclude the paper.

[1]A complete description of RushHour can be found at http://www.thinkfun.com/instructions

Rush Hour Game

Assume that your car is parked in a parking lot and your goal is to reach the exit with your car. The problem is that the path to the exit is blocked by other cars which must first be moved.

The game board consists of a 6×6 grid in which the cells can be occupied by vehicles of different color and type (see Fig. 1). The set of possible positions can be defined as

$$P = \left\{(x,y) \in \mathbb{N}^2 \mid 1 \leq x,y \leq 6\right\}.$$

For a position $p = (x,y) \in P$, $col(p) = x$ denotes the column and $row(p) = y$ denotes the row of p.

The set of vehicles on a Rush Hour board is defined as a set of tuples of a start and end position

$$V = \{(s,e) \mid s,e \in P, row(s) = row(e) \text{ or } col(s) = col(e)\}.$$

There are two types of vehicles which differ only in their length, i.e., by the number of occupied positions. The occupied positions for a vehicle $v = (s,e)$ is the set

$$pos(v) = \{p \in P \mid row(s) \leq row(v) \leq row(e) \wedge col(s) \leq col(v) \leq col(e)\}.$$

There are cars occupying two cells in length and there are trucks occupying three cells in length. Vehicles can be arranged horizontally as well as vertically. The orientation of a vehicle $v = (s,e)$ is horizontal, if $row(s) = row(e)$, otherwise it is vertical. The position of all vehicles determine the occupied board positions.

The exit position is located at the right border of the third row at position $(6,3)$. Thus, the rightmost horizontal car in this row is automatically defined as the exit car. The main goal of the task is to sequentially move the cars such that the exit car can reach the exit, i.e., it can reach the coordinate $(6,3)$. The vehicles can only be moved longitudinally in a forward or backward manner and they are not allowed to leave the grid. Vehicles can only move in the range of free cells. Vehicles may not be moved over or through occupied positions by other cars.

The primary goal of this planning problem is to find a sequence of moves so that the goal condition is reached (satisfying solution). Finding one of the optimal solutions, i.e., a solution with a minimum number of moves, is the secondary goal. The secondary goal is to find one of the optimal solutions, i.e., a solution with a minimum number of moves.

To be able to move the red *car1* in Fig. 1 to the exit, other vehicles (2, 3, 4) that are blocking the route, have to be moved. The complexity of this problem can depend on different factors: (1) The number of vehicles on the grid, i.e., more cars can block each other, but this may also restrict the number of possible moves. (2) The number of moves, i.e., it is harder to find the minimal solution the more moves are necessary as at each step one can deviate from an optimal solution. (3) Counter-intuitive moves, i.e., moves that increase the distance to the goal with regards to optimistic distance measures (e.g. Manhattan distance). (4) The number of branching points for alternative moves where only one specific move

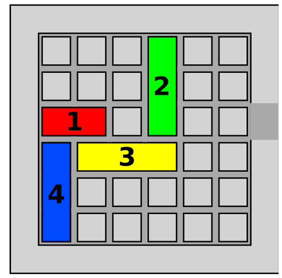

Figure 1: Abstract version of Rush Hour. Cars and trucks are reduced to blocks. The goal is to move the car marked by the number "1" to the exit at the right border of the grid.

leads to the solution. (5) Move dependency, i.e if a previously moved vehicle blocks the movement of other vehicles, which also need to be moved.

Some of these parameters can be calculated offline, i.e without knowing the complete solution (e.g. property 1). Others depend on the actual moves/positions which we denote as online (e.g. property 5).

Cognitive Complexity

Psychologically, reasoning difficulty is measured on a set of problems w.r.t. errors and the time participants need to solve the task. Another often used measure is the relational complexity proposed by Halford et al., (2001; 1998). This complexity measure classifies the problem difficulty by the highest dimensional relation which must be processed simultaneously. Although this complexity measure takes the working memory into account and recognizes the complexity of highly inter-connected tasks – van Rooij et al. (2008) could show that the hypothesis of relational complexity as a measure of difficulty was not confirmed, at least in the case where computational complexity is taken as a measure of difficulty.

Each Rush Hour board can be classified with respect to the branching factor as well as the depth of its optimal solution. While the depth of the solution (or the minimal plan length) will certainly play an important role, the branching factor might not fully reflect human reasoning difficulty, because humans do not always apply all possible operations simultaneously (Anderson, 2000).

Human reasoning is certainly very heuristic driven. In cognitive science a number of heuristics are recognized. The two most well-known are means-end analysis (match the current state to goal state to find the most important difference and eliminate this difference by applying operators (Anderson, 2000, p. 232) and then hill climbing, i.e., choose the operator that transforms the problem state into a state that resembles the goal state more closely than the initial state (Anderson, 2000, p. 228). Further identified heuristics are backward chaining, operator subgoaling, subgoal decomposition, and backup avoidance.

All these characteristics point in the same direction: A cognitive complexity measure must be based on operators – with

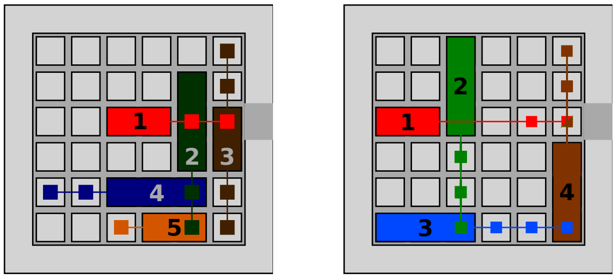

Figure 2: Two example positions and the marked rays for each car. In both pictures *car2* is in the ray of *car1*. To move the red *car1* out each car in the ray has to be moved away. Which is only possible if there is no car blocking in the ray. The ray structure defines the subgoal identification process.

the condition that certain operators must be *cognitively adequate*, i.e., that is the application of this operator must be supported by empirical investigations.

Structural Complexity

As the name implies, a structural complexity measure integrates structural problem characteristics for predicting a problem-inherent difficulty that a planner has to deal with when trying to optimally solve a problem.

We present a complexity measure based on the number and inter-connectedness of task-specific sub-problems. This measure is based on the means-end-analysis heuristic in which a planner successively breaks up a problem into smaller sub-problems (Morris & Ward, 2005). Means-end-analysis seems to be the major strategy applied in the human problem solving process and is the most widely used strategy for modeling human problem solving (Newell & Simon, 1972; Anderson, 2000, 1993). In well-defined problems, sub-problems can be identified by comparing the current state with a desired goal-state and by finding a transformation that reduces the difference between current state and goal state (Miller et al., 1960). We assume that for spatial transformation problems like Rush Hour, the euclidean distance between the current position of a vehicle and its goal position is an appropriate measure for the goal distance.

Based on this, sequential sub-problems can be represented as a graph, with the sub-problems as nodes which are connected by directed edges to represent a sequential dependency. This graph can contain cycles because problem chains might occur that reference back to an earlier sub-problem. For example in Fig. 1 the exit *car1* is blocked by *car2*, but to solve this blocking, *car2* has to move down. This move is blocked by *car3* which is blocked by *car4*. To move *car4* such that it does not block *car3* requires the exit *car1* to move, i.e., we have a cyclic reference back to the beginning.

In Rush Hour, sub-problems are defined based on the blocking of desired goal positions. For example, the initial goal requires the red car to be at position $((5,3),(6,3))$. To be able to reach this position, the goal position itself as well as all positions in between need to be free. Thus, all vehicles in between the current position and the goal position are sub-problems. The set of cells that need to be freed to move a vehicle to its goal position is defined by the *blocking ray* of the vehicle (see Fig. 2).

Definition 1 *The* ray *of a vehicle* $v = (s,e) \in V$ *with reference position* $p = (p_x, p_y) \in P$ *is the set of positions*

$$R_v^p = \{(x,y) \in P \,|y = row(s), col(e) < x \leq p_x \vee \\ p_x \leq x < col(s)\}$$

if v is horizontal and

$$R_v^p = \{(x,y) \in P \,|x = col(s), row(e) < y \leq p_y \vee \\ p_y \leq x < row(s)\}$$

if the blocked vehicle v is vertical.

Now we can define blocking cars as the set of cars which occupy a position on a cars' blocking ray as follows:

Definition 2 *The blocking cars for a car* $v \in V$ *regarding a reference position* $p \in P$*, are defined as the set*

$$B_v^p = \{c \in V \mid pos(c) \cap R_v^p \neq \emptyset\}.$$

If the blocking vehicles for a car regarding its desired goal position are known, the successive sub-problems can be derived directly. For each blocking vehicle, possible positions that do not block the current goal position are new goal positions for the successive sub-problem. The generation of new sub-problems then continues as long as there are successive sub-problems or as long as the new sub-problem does not involve a vehicle that was already considered in the sub-problem chain back to the initial problem.

The successive sub-problem generation can be used to convert a Rush Hour board into a directed graph $G = (N,E)$ representing the degree of interlacing between the vehicles (see Fig. 3). The node set $N \subseteq V$ is the set of vehicles which are necessary for the solution. The directed edges represent the blocking relations. These are defined as:

$$E = \{(v_1, v_2) \in N^2 \mid v_2 \in B_{v_1}^p\}$$

i.e., two nodes are connected, if the car v_2 is in the set of blocking cars of car v_2. The blocking cars for a car can only be determined if a goal position p is known. These goal positions are determined successively beginning from the initial goal.

The generation of the structural graph is a recursive process beginning with the exit car v_e having the goal position $p = (6,3)$. For each blocking car $v \in R_{v_e}^p$ a new edge (v_e, v) is added to the graph. The definition of sub-problems could lead to infinite loops if the problem contained cycles. Therefore, if the new sub-problems contain already visited nodes we use a special edges marking this back reference. A node n, representing a newly generated sub-problem of a predecessor node p, is called *visited* if G contains a path from n to p that is not a direct connection, i.e., it is not a back reference edge. Then, a new edge (p,n) is inserted and the search is stopped.

If the back reference connections are disregarded, the graph is directed and acyclic, i.e., it is a tree with the exit car as root node. Thus, we can assign each node a depth regarding the tree representation of the graph. As mentioned above, the structural complexity measure represents the number and inter-connectedness of the sub-problems that are necessary to solve the problem.

The numerical value for the structural complexity is determined by the problem graph and its reduced tree representation. For a node v with successor nodes $s_1,\dots,s_n, n \in \mathbb{N}$, with back references from "deeper" nodes $b_1,\dots,b_m, m \in \mathbb{N}$, the complexity of v is defined as

$$c(v) = \underbrace{\sum_{i=1}^{n} [c(s_i)+1]}_{\text{sub-problems}} + \underbrace{\sum_{i=1}^{m} depth(b_i)}_{\text{backreferences}} .$$

Leaves are either not blocked, or are part of a cycle. The complexity of such nodes is counted as 0. A node's complexity thus results from the number and complexity of its successor nodes.

Figure 3 shows a graph conversion for a sample Rush Hour task. The complexity of each node is given in the node caption. The exit car (*car1*) is blocked by *car2* and *car3*. They are, therefore, inserted into the graph as direct successor nodes of the root node. On the next level, *car4* is blocked by *car5* which in turn is blocked by the exit car so that a back reference edge goes from *car5* to *car1*. On the final level, *car6* blokcs the movement of *car2* and *car3*, but *car6* itself is not hindered from freeing either of these cars. Leaving *car6* as the last node added to graph.

The resulting node complexities are computed in a bottom-up manner. First, *car6* and *car5* are leaves in the tree representation of the graph and thus have a complexity of 0. The complexities of the nodes *car2*, *car3* and *car4* are calculated by adding one to the complexity of each successor node and then adding the complexities together. For the root node *car1*, the final task complexity results from the complexity of the two successor nodes as well as from the depth of the back-reference of *car5*. The successor node complexities add up to 7 and the depth of the back-reference is 3 and thus, the overall task complexity is 10.

Cognitive Complexity and Empirical Difficulty

We conducted a behavioral experiment to test if participants could find the optimal number of moving steps and if not, why and what were they doing instead.

Participants, Material, and Task. Twenty participants (8 male, 12 female, mean = 24,8 years) processed 21 tasks from

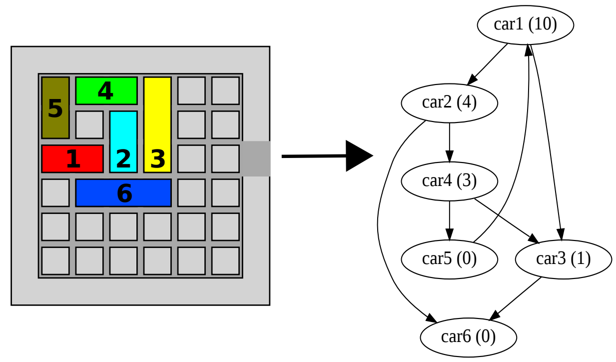

Figure 3: Rush Hour board and its structure graph. The complexities of each node are given in the trailing parentheses. The total complexity of the task results from the complexity of the root node. The example given here has a total complexity of 10.

the "Junior edition" of the Rush Hour game. Tasks were selected with respect to following aspects: existing classification of the tasks (beginner, intermediate, and advanced), the minimum number of needed to reach the solution, and number of additional moves for the exit car (one move at the end to the right to the exit, two separate moves to the right, and one counter intuitive move to the left and afterwards a separate move to the right). The participants solved all tasks on the computer. Behavioral parameters like solution length, moves made, and moving time for each step were recorded.

Results. For some of the Rush Hour tasks we show exemplary statistical and structural complexity results. The accuracy of different difficult tasks (easy, moderate, high) are shown in Table 1. For a better understanding of the human planning abilities it is especially important to explore the tasks which were solved but not with the optimal solution length.

Table 1: Accuracy for six selected tasks (easy 6, 13, moderate 9, 10, hard, 27, 29) in percent for 20 participants (NS = not solved, SN = solved, but not optimal, SO = solved optimal).

Problem	6	13	9	10	27	29
NS	0%	0%	0%	0%	10%	20%
SN	5%	15%	80%	100%	80%	80%
SO	95%	85%	20%	0%	10%	0%

The increasing mean move difference from optimum as well as the increasing standard deviation in relation to difficulty of the task is displayed in Table 2. Almost all statistical parameters reflect the increasing difficulty of the tasks. Note, these parameters can better explain the difficulty difference for humans between moderate and hard tasks.

The over-all results in Table 3 indicate that the difficulty classification by means of statistical parameters (mean, stan-

Table 2: The descriptive statistic shows the move difference from the optimal number of moves for correct answered easy (6, 13), moderate (9, 10), and hard (27, 29) tasks.

Task	n/all	Mean	SD	Med	Min	Max
6	20/20	0.10	0.45	0	0	2
13	20/20	0.30	0.80	0	0	3
9	20/20	4.45	4.20	3	0	13
10	20/20	6.05	5.12	4	1	17
27	18/20	11.00	11.21	8.5	0	37
29	16/20	10.56	8.45	10.5	1	32

dard deviation) is possible. A more profound analysis of the online parameters would go beyond the scope of this paper.

Table 3: Over-all results for all tasks can classify empirical difficulty. Only the hard tasks have a substantial unsolved rate and also a higher mean moving steps difference.

Difficulty	optimally solved	solved	not solved	mean move diff (SD)
easy	$\geq 85\%$	100%	0%	< 1 (< 1)
moderate	$\leq 20\%$	$\geq 96\%$	$\leq 4\%$	$4-6$ ($4-6$)
hard	$\leq 10\%$	$\geq 62\%$	$\leq 38\%$	> 10 (> 7)

Likewise, the offline parameters such as the structure complexity in Table 4 fit the calculated online difficulty. The structural complexity calculation, especially with consideration of back-references is able to predict the empirically determined difficulty.

Table 4: Structural complexity with/without back-references (distance to the reference node is weighted/not considered).

Task	with back-reference	w/o back-reference
6/13	3/5	3/5
9/10	22/16	18/13
27/29	48/30	24/18

Critical offline parameters were calculated (see Table 5) and correlated with online parameters which were obtained from the actual moving track. To capture the empirical deviations from the formal structure complexity measure we require a measure to describe the goodness of the solution.

F1: $MEAN(move_n / optimal\ moves)$

F2: $\sum_n (move_n - optimal\ moves)$

F3: $log_2(\sum_n (move_n - optimal\ moves))$

Minimal Solution Length (MSL) denotes the minimum (optimal) number of moves for the solution. *Structural Complexity* (SC) denotes the complexity with back references which were calculated from the starting point of a task. Sum of Structural Complexity (SSC) denotes the mean cumulated structural complexity of all optimal solutions. Thus, the structural complexity for each step of each optimal solution was calculated and summarized and the mean over all optimal solutions

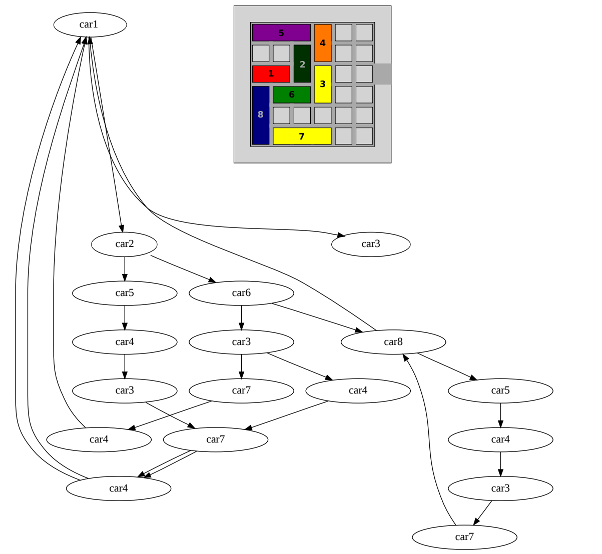

Figure 4: Board setting and structural graph with back references for task 27 of the Junior edition was generated by a computer program for analyzing Rush Hour tasks. The nodes contain the car and the edges denotes the directed blocking relation between the two connected vehicles. This graph shows a complex network of blocking relations and cyclic blocking chains.

was built. Sum Movable Cars (SMC) denotes the cumulated sum of movable vehicles of all optimal solutions. This means that the possible movable vehicles for each step of each optimal solution was calculated and summarized and the mean over all optimal solutions was built. All values were transformed with log_2 and can be described as offline parameters. The formulas below are used to calculate the results from the empirical online parameters.

Table 5: The table indicates correlations between different parameters: Minimal Solution Length (MSL): minimal number of moves for solution; Structural Complexity (SC): complexity with cycles calculated from the starting point of a task; Sum Structural Complexity (SSC): mean cumulated structural complexity of all optimal solutions; Sum Movable Cars (SMC): number of movable cars. All values were transformed with log_2. ($^{*}\alpha \leq 0.05$; $^{**}\alpha \leq 0.01$; $^{***}\alpha \leq 0.001$)

	MSL	SC	SSC	SMC
F1	$r = .47$	$r = .06$	$r = .37$	$r = .41$
F2	$r = .77$***	$r = .44$	$r = .64$**	$r = .60$**
F3	$r = .66$**	$r = .51$*	$r = .66$**	$r = .62$**

General Discussion

We investigated formal and empirical properties for a cognitive complexity measure designed to be able to classify planning problems w.r.t. factors of human reasoning difficulty. This investigation lead to the development of a structural complexity measure which we applied to the planning task Rush Hour. The structural complexity might be an essential part of cognitive complexity as it reflects the principle of means-end analysis (MEA) and difference reduction, i.e., a greedy strategy to minimize a goal distance measure. A concrete example of this general strategy is the interlaced vehicle blockades in the tasks' structural graph conversion. More complex graph structures indicate more complex subgoal decompositions for solving the task. This aspect is represented in the structural complexity measure. We assume that this complexity measure is applicable to other domains as well, since means-end-analysis seems to be the premier human problem solving method (Anderson, 1993).

The statistical analysis of Rush Hour tasks indicates that the empirical reasoning difficulty significantly correlates with the occurrence of move chains (of cars blocking each other). Compared to simpler tasks, more difficult tasks contain more subgoals to be solved and contain cyclic blocking structures (cf. Fig. 3) in their associated graph. Based on the significant correlations, the structural task complexity measure seems to be a good indicator for our hypothesis of the empirical reasoning difficulty.

A previous computational analysis revealed that planning depth alone does not capture the "cognitive" difficulty in solving planning tasks like Rush Hour. Of course, there is a correlation, but a higher number of necessary moves also results in a greater chance (depending on each branching point) of deviating from the optimal plan length. A deviation from the optimal number of moves for solving the task likely reduces the cognitive planning effort drastically but might not consider essential information about future states. Therefore, it is important to analyze the problems that were used in the experiments regarding a systematic deviation from the optimal solution length. With a detailed analysis of branching points, the planning depth as well as applied heuristics could be identified, which gave additional information about the human planning quality.

In neuropsychology planning tasks like Tower of London (ToL) are used to classify prefrontal lesions (Shallice, 1982; Unterrainer et al., 2004). However, compared to Rush Hour the task structure of ToL is too simple w.r.t. applicable heuristics to fully classify human planning abilities. The number of optimal solved tasks of Rush Hour compared to optimal solved tasks in ToL had a significant correlation ($r = .55, p < .01, n = 20$). Rush Hour is computationally more complex than ToL due to the higher number of possible moves and heuristics reflected by the associated tree conversion.

A detailed move analysis might reveal, in what kind of problem states humans will probably deviate from optimal solutions. We are currently working on further measures to capture these aspects. Our goal is to classify human players that share similar planning characteristics and apply similar heuristics.

Acknowledgements

This research was supported by the DFG (German National Research Foundation) in the Transregional Collaborative Research Center, SFB/TR 8 within project R8-[CSPACE].

References

Anderson, J. R. (1993). Problem solving and learning. *American Psychologist*, *48*, 35–44.

Anderson, J. R. (2000). *Cognitive psychology and its implications* (3rd ed.). New York: Worth.

Ellis, S., & Siegler, R. S. (1994). Development of problem solving. In R. J. Sternberg (Ed.), (2nd ed., pp. 333–367). San Diego: Academic Press.

Flake, G. W., & Baum, E. B. (2002). Rush hour is pspace-complete, or "why you should generously tip parking lot attendants". *Theor. Comput. Sci.*, *270*(1-2), 895–911.

Halford, G. S., Phillips, S., & Wilson, W. H. (2001). Processing capacity limits are not explained by storage limits. *Behavioral and Brain Sciences*, *24*(1), 123–124.

Halford, G. S., Wilson, W. H., & Phillips, S. (1998). Processing capacity defined by relational complexity: Implications for comparative, developmental, and cognitive psychology. *Behavioral and Brain Sciences*, *21*(6), 803–831.

Helmert, M. (2008). *Understanding planning tasks: Domain complexity and heuristic decomposition* (Vol. 4929). Berlin Heidelberg: Springer.

Kotovsky, K., Hayes, J. R., & Simon, H. A. (1985). Why are some problems hard? evidence from tower of hanoi. *Cognitive Psychology*, *17*, 248–294.

Miller, G., Galanter, E., & Pribram, K. (1960). *Plans and the structure of behavior*. New York: Holt, Rinehart and Winston.

Morris, R., & Ward, G. (2005). *The cognitive psychology of planning* (K. J. Gilhooly, Ed.). New York: Psychology Press.

Newell, A., & Simon, H. A. (1972). *Human Problem Solving*. Upper Saddle River, NJ, USA: Prentice-Hall.

Papadimitriou, C. M. (1994). *Computational complexity*. Reading, Massachusetts: Addison-Wesley.

Rooij, I. van, Evans, P., Müller, M., Gedge, J., & Wareham, T. (2008). Identifying sources of intractability in cognitive models: An illustration using analogical structure mapping. In V. Sloutsky, B. Love, & K. McRae (Eds.), *Proc. of the 30th annual conference of the cognitive science society* (pp. 915–920). Texas: Cognitive Science Society.

Russell, S. J., & Norvig, P. (2003). *Artificial intelligence: A modern approach (second edition)* (2nd edition ed.). Englewood Cliffs, NJ: Prentice Hall.

Shallice, T. (1982). Specific impairments of planning. *Philosophical Transactions of the Royal Society London B: Biological Sciences*, 199–209.

Unterrainer, J. M., Rahm, B., Kaller, C. P., Ruff, C., Spreer, J., Krause, B. J., et al. (2004). When planning fails: individual differences and error-related brain activity in problem solving. *Cerebral Cortex*, *14*, 1390–1397.