

# A production system model for human problem solving

F. Schmalhofer<sup>1</sup> and P.G. Polson<sup>2</sup>

<sup>1</sup>Psychologisches Institut der Universität Freiburg, Niemensstr. 10, D-7800 Freiburg i.Br., FRG

<sup>2</sup>Department of Psychology, University of Colorado, Boulder, Colorado 80309, USA

**Summary.** A new production system model was developed for a class of transformation problems, based upon the results of an experiment which evaluated the psychological validity of a noticing order in problem solving. This model specifies how general problem solving heuristics interact with domain knowledge and how possible actions are assembled from externally presented information. The production system may conceptually be divided into three groups: move generation, move evaluation, and move execution productions. The simulation model was evaluated by a second experiment. With a common set of parameter values, good fits were obtained between the predicted and observed data. The significance of individual productions was assessed by running simulations with individual productions deleted. While previous simulations have described subjects' behavior on a single problem, the present model successfully predicted human problem solving in three structurally different problems. The proposed production system thus gives a detailed account of the procedural knowledge that subjects use in transformation problems.

The early research on problem solving (Duncker, 1935) has already specified how humans use heuristic search processes to overcome the discrepancy between the given and the desired state of affairs in a problem solving task. With the advent of computers and the use of computer simulations in cognitive psychology it became possible to demonstrate that problems can indeed be solved by goal-driven search processes such as the means-ends heuristic (Ernst & Newell, 1969).

However, a computer simulation program often contains a number of additional specifications which are needed for purely technical reasons. For example, early computer simulations in psychology were often programmed in compiled languages, which usually require an explicit and complete specification of all structures and operations before any segment of the program can be executed (Pratt, 1975). Given such a simulation environment (Kieras, 1985) it is quite reasonable to explicitly specify all the operators which can be used for solving a problem before the first step of problem solving is executed in the simulation program. Whether the specification of a list of operators carries any psychological significance or validity is a different issue. Nevertheless, as simulation programs were more seriously assessed as psychological process models, the assumption that problem solvers have a list of possible moves stored in memory

was examined in psychological experiments (Atwood, Masson, & Polson, 1980), so that the assumption of a list of possible moves in the problem solver's memory, which has been termed noticing order, can no longer be seen as a purely technical assumption (Weisberg, 1980, p. 267). A noticing order implies that in every single state of the problem, the problem solver considers possible moves in the exact same order.

Since heuristic strategies are rather flexible (Newell & Simon, 1972) and may even adapt to the particular givens of the currently displayed (external) information (Schmalhofer, Albert, Aschenbrenner, & Gertzen, 1986), problem solving operators may instead be derived from the externally displayed state of the problem than from a fixed and internally represented noticing order, which has been postulated by Jeffries, Polson, Razran, & Atwood (1977).

In order to examine the psychological validity of a noticing order, an experiment was performed in which subjects had to solve river-crossing tasks with or without an assumed noticing order being explicitly displayed in each problem state. The problem solving tasks used in this study will be described in the next section of the paper. Based upon the results of the first experiment, a new production system model was developed which specifies how possible moves are generated from the givens of a problem state. In addition to the previously postulated problem solving heuristics such as means-ends analysis (Newell & Simon, 1972) and the memorization of moves (e.g., Atwood & Polson, 1976) the proposed model also postulates that humans utilize task-specific information. The simulation model was evaluated by a second experiment for which human subjects' move behavior was predicted by the model. The significance of some key productions for the overall behavior of the production system model was assessed by running simulations with these critical productions deleted.

## River-crossing task

River-crossing problems are distinguished from the Tower of Hanoi (Simon, 1975) and other move problems by their extremely irregular problem structure, which makes it rather unlikely that a problem solver can perform any multi-move planning. Contrary to the Tower of Hanoi problem, where a recursive evaluation helps to define subgoals and to find the optimal solution path (Anzai & Simon, 1979; Simon, 1975), no such analysis is readily available in river-crossing problems. Consequently, these problems are particularly suited for investigating human problem solving strategies for task environ-

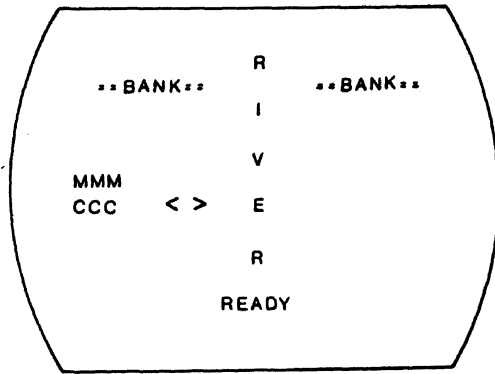


Fig. 1. Sample display from the CRT

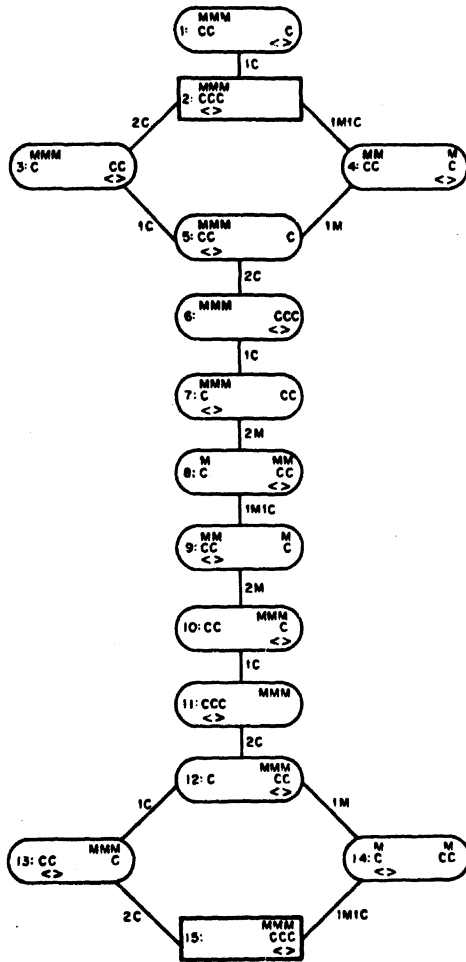


Fig. 2. Problem graph of the (3,2) task

ments which involve only little or no understanding of the problem characteristics.

In the start state of the river-crossing problem there are  $n$  missionaries and  $n$  cannibals on the left or start bank of a river. In order to continue their journey, these travellers must cross the river in a rowboat that can only hold  $k$  of them. Since the cannibals would attack the missionaries if they were in the majority, the cannibals must never outnumber the missionaries. In order for the boat to travel from one side of the river to the other, at least one person must be rowing the boat. In order to solve this problem a sequence of moves must be determined by which the missionaries and the cannibals are transferred to the right or goal bank. We will let  $(n, k)$  denote a problem with  $n$  missionaries,  $n$  cannibals and a boat that can hold at most  $k$

passengers, where  $k$  is smaller than  $n$ . Figure 1 shows the display of the start state of the (3,2) problem with 3 missionaries, 3 cannibals and a boat for 2 passengers.

Figure 2 shows the problem graph of the (3,2) problem. Every possible legal state of the problem is represented by a box. Legal moves are shown as connecting lines between two problem states. The rectangular box at the top which is labeled as state 2 denotes the start state. The rectangular box at the bottom which is labelled state 15 is the goal state. Since for the overwhelming majority of problem states of the (3,2) problem there are only two possible moves, a forward and a backward move, the (3,2) problem has a basically linear problem graph.

The (5,3) problem with 5 missionaries, 5 cannibals and a boat for 3 passengers has a quite different structure. As seen from Fig. 3 there is a bushy part at the top and the bottom and a very small linear part in the middle of the problem graph. Still different from both of these problems is the (4,3) task (see Fig. 4) which allows two completely independent solution paths that do not share a single problem state. These three problems were selected for the current study because the three structurally different problems with linear, bushy, and two parallel paths in the problem graphs should provide a rather general test of the proposed simulation model.

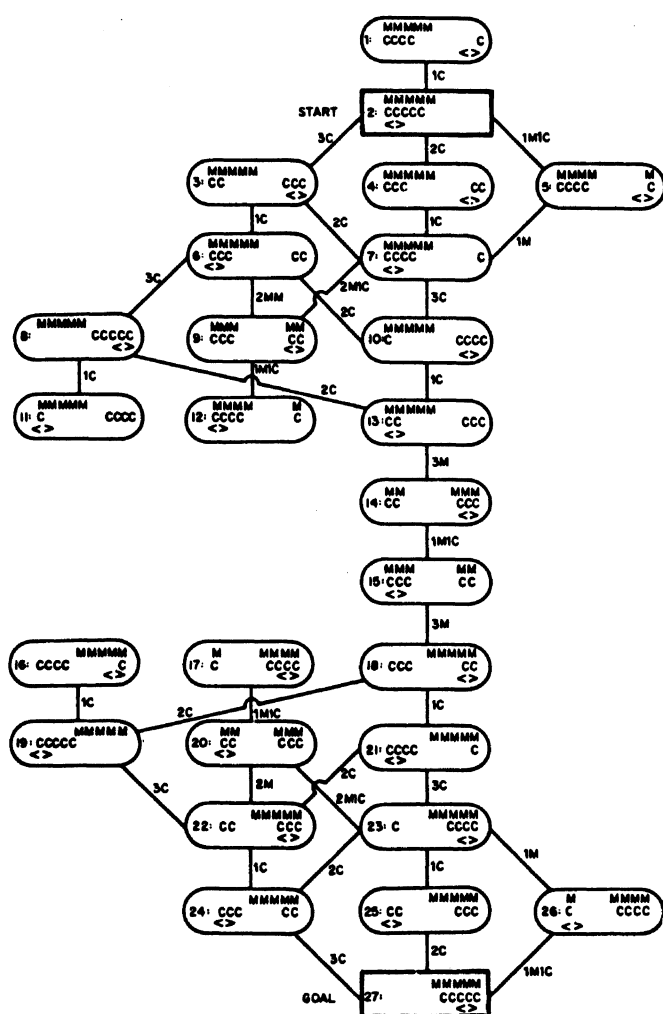
It has been proposed that humans solve problems by applying general heuristics similar to the General Problem Solver (GPS) (Ernst & Newell, 1969), which was capable of solving a variety of different problems by the means-ends strategy. Although Greeno (1974) has demonstrated that GPS is not directly a model of human problem solving, cognitive psychologists often assume that human problem solving is GPS-like. Such GPS-like models have been developed by Atwood and Polson (1976), Jeffries et al. (1977), and Simon and Reed (1976).

Although the Jeffries et al. model predicted subjects' behavior in the (3,2) problem, the subjects' performance in the (5,3) problem could not be accounted for by this model without introducing baroque ad hoc assumptions into the simulation (Polson & Jeffries, 1982, p. 397). This failure may possibly be due to the noticing order which is postulated by the Jeffries et al. model. In order to determine the empirical validity of the noticing order an experiment was performed.

### Experiment 1

Three conditions were employed in the experiment. In the two experimental conditions all possible moves and the states resulting from these moves were shown, however, in a different order to the subjects together with the presentation of the current problem state. In the control condition only the present state of the problem was shown without any moves being displayed. If humans do indeed keep a noticing order in memory and mentally calculate and evaluate the resulting state before they take the corresponding move, the experimental problems should be easier, because the experimental display presents information which must otherwise be stored in working memory (Atwood et al., 1980).

If subjects evaluated resulting states, problem solving should thus be better in the experimental than in the control condition, because additional processing resources become available in working memory. In one of the experimental conditions the list of possible moves was arranged by the noticing order which is postulated by the Jeffries et al. model. Thus this experiment will in particular test the empirical validity of the noticing order which has been postulated by the Jeffries et al. model.



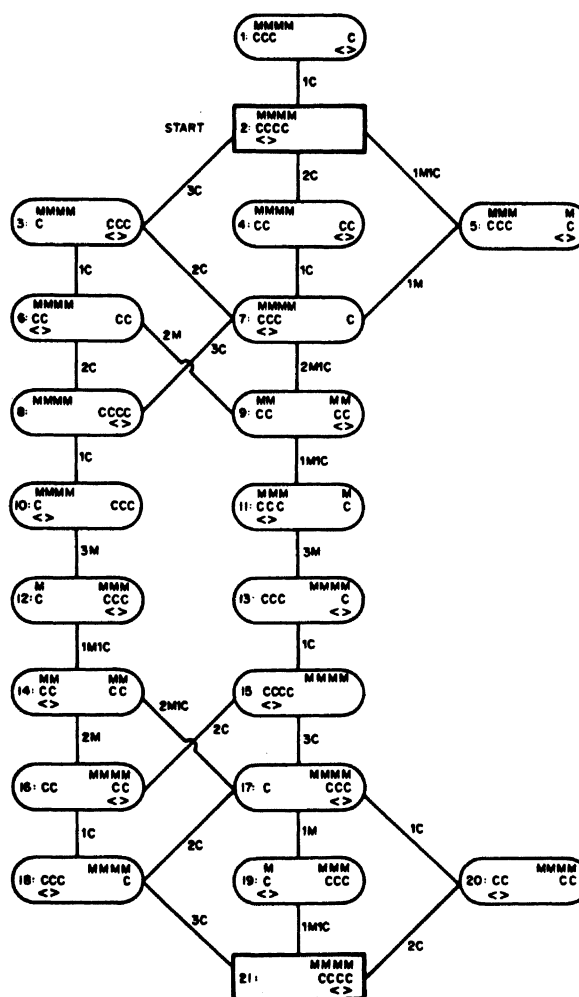
**Fig. 3.** Problem graph of the (5, 3) task

## Method

**Subjects.** 120 subjects were recruited from introductory psychology classes at the University of Colorado. They received class credit for their participation.

**Apparatus.** The experiment was controlled by a Xerox Sigma 3 computer. The problems were presented to the subjects on a IV Phase System CRT display terminal by a general-purpose program for problem solving experimentation (Jeffries & Karat, 1979). The subject responded by pressing buttons. There were five buttons labelled from left to right: MISSIONARY, CANNIBAL, SEE IT, DO IT and ERASE<sup>1</sup>. The procedure was subject paced, and an independent sequence of events was presented to each subject.

**Procedure.** Each subject was first given general instructions about the experiment. Detailed instructions concerning both the problem and the method of responding were presented on the CRT, which the subjects read at their own pace. The instructions for the three subject groups were identical except for whether a noticing order would be displayed and which noticing order would be displayed. In the noticing order condi-



**Fig. 4.** Problem graph of the (4, 3) task

tions all possible moves from the given state were displayed on the right half of the screen together with the corresponding resulting states. The left half of the screen displayed the current problem state in a similar way as shown in Figure 1.

There were two experimental groups and a control with 40 subjects each for which noticing order was displayed. For experimental Group 1 a missionary first means-ends order (MMM, MMC, MC, CCC, MC, MM, CC, M, C) was used. For experimental Group 2 a noticing order which is also a means-ends order, but clearly different from the order which has been assumed by Jeffres et al. was displayed (CCC, CCM, CMM, MMM, CM, CC, MM, C, M). In all other respects the statement and presentation of the problems were identical.

Once a subject had started the problem, he or she could not return to the instructions. However, they could call the experimenter if they had any questions. The subjects entered a move by pressing the MISSIONARY and CANNIBAL buttons once for each traveller they wanted to load into the boat. They then pressed SEE IT and the creatures were loaded into the boat but were not yet ferried to the other side of the river. The move was erased if the subject pressed the ERASE button. If the subject pressed the DO IT button, the move was recorded as a legal or illegal move. Illegal moves are moves violating the restrictions of the problem description. Appropriate messages were given for illegal moves and the problem continued with the latest legal state of the problem. The subject worked on the problem until he or she solved it, made 100 legal moves without solving it, or after one hour had elapsed.

**Table 1.** Means and standard deviations of the (5,3) problem for legal, illegal, and erased moves

Moves	Experimental group 1		Experimental group 2		Control group	
Legal						
Mean	27.85	(29.74) <sup>a</sup>	27.88	(27.23) <sup>a</sup>	27.10	(28.90) <sup>a</sup>
SD	14.05	(16.18) <sup>a</sup>	14.33	(14.72) <sup>a</sup>	16.49	(14.78) <sup>a</sup>
Illegal						
Mean	6.70	(6.64) <sup>a</sup>	4.28	(5.48) <sup>a</sup>	5.75	(6.45) <sup>a</sup>
SD	8.77	(5.54) <sup>a</sup>	4.17	(4.70) <sup>a</sup>	12.06	(5.06) <sup>a</sup>
Erased						
Mean	3.53		4.39		4.85	
SD	3.54		3.92		4.74	
Errors						
Mean	2.40		2.15		1.40	
SD	2.84		2.90		2.52	

<sup>a</sup> Predictions computed by the production system model, using the parameters shown in Table 6; (5,3)-set

## Results

Means and standard deviations for legal, illegal, erased and error moves are shown in Table 1. For every subject a 15-item and an 11-item vector were produced for legal and illegal moves, respectively. The 15-item vector contains the number of times states 2, 3, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 18, 21, and 23 were visited by every subject. Similarly, the 11-item vector contains the number of times illegal moves were taken from states 2, 5, 6, 7, 9, 11, 12, 13, 14, 15, and 18. Other states were not included in this analysis because the respective numbers were zero or close to zero. Grant's repeated measures analysis of variance found neither significant differences in overall performance nor significant differences for legal or illegal moves. There were no differences in the move-profiles (all  $F < 1$ ). The correlation coefficients of the move-transition frequencies between the control and the two experimental subject groups were both greater than .95. In order to determine whether an externally displayed noticing order leads to faster move selections, the above analysis was repeated with average move latencies replacing move selections. Again, no significant differences were found: (all  $F < 1$ ).

## Discussion

The experimental results did not provide any empirical support for the notion of a noticing order. Also the move latencies were identical for the three groups. This clearly shows that subjects ignored the information on the right half of the screen altogether. Using the same experimental procedure Atwood et al. (1980) demonstrated significant effects of externally displayed moves for water jug problems. The present results are therefore substantive and not due to the experimental procedure. Based on the economy principles of the human information processor it is concluded that ignoring the possible-move display was more efficient for the subject than processing it. This, however, would imply that in river-crossing tasks humans neither use a noticing order, nor do they mentally calculate the resulting state before deciding whether they will perform a certain move. Based upon these results a new simulation model was developed.

## The model

The present model assumes that subjects use domain specific knowledge in addition to general problem solving heuristics. Presumably a subject constructs procedural knowledge from the problem instructions and previously acquired real-world knowledge. Usually a subject will not immediately integrate the different knowledge fragments into a consistent or coherent knowledge structure. Thus, since no coherent solution plan is developed, different knowledge segments may be in disarray with one another.

In the case of most river-crossing tasks the outnumbering restriction, which is ideally satisfied by moving all travellers in pairs, is in conflict with the goal to transfer all travellers to the right bank of the river. The balance strategy, which attempts to keep the number of missionaries and cannibals balanced on each side of the river, tries to satisfy the outnumbering restriction. The means-ends strategy pursues the goal of transferring all missionaries and cannibals to the right bank of the river. Since the balance and the means-ends strategies are not integrated into a coherent mental representation, a probabilistic mechanism determines which strategy is applied at any given time. Before describing the single productions, some general characteristics of the proposed model will be mentioned.

A memory model with long-term, working and sensory memory is assumed. Long-term memory stores the characteristics of the particular river-crossing problem. During the problem solving process some problem states will also be stored in long-term memory. The information stored in the three memory stores is shown in Table 2. Working memory stores the move most previously taken and a list of no more than three moves which the production system generates and operates upon during the move-selection process. Thus, productions may store information in working memory and also test the contents of working memory. After two illegal moves have occurred in a row a flag is set in working memory. Setting this flag guarantees the application of the illegal move filter. Finally, sensory memory is assumed to contain the information which is externally displayed to the subject during any state of the problem.

The simulation model is a probabilistic, basically flat production system where the first production whose condition matches

**Table 2.** Data used by the production system

Information in LTM	
(BOATSIZE)	
(GOAL)	Characteristics of particular river-crossing problem
(PAIRNUM)	
(LTMSTATE)	Memory of visited states
Information in WM	
(PREMOVE)	Previous move taken
(MOVLIST)	List of proposed moves
(LGLTY)	Number of illegal moves after last legal move
Information in external and sensory memory	
(DISPLAY)	Description of display
(CODE)	Encoding of display

will fire. The productions may conceptually be subdivided into three groups. The proposed productions generate possible moves for the current state of the problem and test whether a move is stored in working memory. If the proposed productions fire, they may store up to three potential moves in working memory. The second group of productions are the evaluate productions which fire when a proposed move is stored in working memory. The evaluate productions either approve the proposed move or they delete the proposed move from working memory. Finally, the execute productions apply the illegal move filter and a long-term memory test and then execute the approved move. The cycle of proposing, evaluating, and executing moves continues until the problem is solved. The model consists of eleven productions, six parameters and an illegal move filter. The parameters represent the probabilities that a certain production fires. There are probabilistic as well as deterministic productions. The eleven productions may be individually described in the following way.

#### *Move generation process*

The move generation consists of two productions, a balance production (GE1) and a traveller sampling production (GE2), which are described in Table 3. The balance production tries to

**Table 3.** The productions of the move generation process

<b>GE1: Balance production</b>	<b>Parameter: P1</b>
If (no move in WM, and difference between Ms and Cs of <DISPLAY> smaller than <BOATSIZE>)	
----->	(generate moves to balanced states and store these moves in WM)
<b>GE2: Traveller-sampling production</b>	
If (no move stored in WM)	
----->	(generate means-ends moves by sampling travellers from <CODE>, store generated moves in WM)

generate a move which leads to a balanced state of missionary-cannibal pairs on either side of the river. These moves are easily generated by moving all those travellers who do not have a partner. The traveller-sampling production uses a means-ends driven sampling procedure to generate moves. Travellers are sampled without replacement from that side of the river where the boat is. Many travellers are moved when the boat is on the start bank and few if the boat is on the goal bank. Both the balance and the traveller sampling production store their generated moves in working memory.

#### *Move evaluation process*

The proposed move will then be evaluated by the move evaluation process which has three levels. Level zero move evaluation (EV1) occurs with probability P5 which is typically small. In all other cases level one move evaluation (EV2, EV4, EV5) will be conducted. An additional level two move evaluation (EV6, EV7) may occur with probability P6. The seven move evaluation productions are specified in Table 4.

The guaranteed approval production (EV1) unconditionally approves any move. This production was included in order to account for moves which subjects take very infrequently, such as visits to a state number one. The level one move evaluation consists of production EV2 which is a task specific production. This production rejects moves to state number 1 and models the

**Table 4.** The productions of the move evaluation process

<b>Level 0 move evaluation</b>	<b>Parameter: P5</b>
<b>EV1: Guaranteed approval production</b>	
If (proposed-move in WM)	
----->	(approve this move)
<b>Level 1 move evaluation</b>	
<b>EV2: Nonsense-move-rejection production</b>	
If (present state is start state & unapproved move in WM contains only one traveller)	
----->	(reject this move by deleting it from WM)
<b>EV3: Avoid-backward-move production</b>	<b>Parameter: P7</b>
If (unapproved move in WM is equal to the previous move)	
----->	(reject this move by deleting it from WM)
<b>EV4: Avoid-start-state production</b>	<b>Parameter: P7</b>
If (unapproved move clears goal bank)	
----->	(reject this move by deleting it from WM)
<b>EV5: Accept-goal-state production</b>	
If (unapproved move clears start bank)	
----->	(approve this move)
<b>Level 2 move evaluation</b>	<b>Parameter: P6</b>
<b>EV6: Subgoal-protection production</b>	
If (all Ms on goal bank and unapproved move suggests to transfer Ms)	
----->	(if boat is on start bank replace all Ms else delete all Ms from move)
<b>EV7: Means-ends-evaluation production</b>	
If (unapproved move in WM violates means-ends criteria)	
----->	(reject this move by deleting it from WM)

**Table 5.** The productions of the move execution process

<b>EX1: Avoid-memorized-state production</b>	<b>Parameter: P3</b>
If (approved move in WM and its resulting state is stored in LTM)	
----->	(reject this move by deleting it from WM)
<b>EX2: Illegal move filter</b>	<b>Parameter: P4</b>
If (illegal move filter does not detect illegal move)	
----->	(then execute this move and delete it from WM)

subject's understanding that if only one cannibal is brought to the goal bank the cannibal must row the boat back to the start bank. Subjects obviously have enough task specific knowledge in order to recognize this move as a nonsensible action.

Productions EV3, EV4, and EV5 specify what is usually called general problem solving heuristics for transformation problems. Productions EV3 and EV4 try to avoid moves back to the previous state and to the start state. Finally, production EV5 recognizes the goal state as the solution to the problem. Another example of task-specific procedural knowledge is production EV6. This production indicates that subjects recognize all missionaries on the right side of the river as a subgoal which should be maintained. However, this subgoal is only recognized after it has been accomplished. Thus, production EV6 will not fire unless all missionaries are at the goal bank already. Production EV7 rejects moves which transfer *k* travellers to the start bank.

#### *Move execution process*

The move execution process consists of two productions which are shown in Table 5. The first production (EX1) eliminates moves to states which have been stored in LTM. Production EX2 applies the illegal move filter and, when this test does not detect an illegal move, executes this move. The move execution

**Table 6.** Three sets of best fitting parameters for the (3,2), (5,3) and both problems combined

Production	Problem		
	(3,2)	(5,3)	General
P1 Balance	0.30	0.35	0.35
P2 Sample	0.70	0.50	0.56
P3 LTM	0.30	0.30	0.30
P4 Legality	0.85	0.90	0.92
P5	0.05	0.05	0.04
P6 Deep evaluation	0.40	0.40	0.43
P7 Reverse	0.10	0.10	0.10

process also stores the previously visited state with probability P3 in long-term memory. The Appendix presents some more detailed information about the simulation program.

The simulation model which has been developed by using the experimental data collected in Experiment 1 was to be tested by an experiment with (3,2), (5,3), and (4,3) problems. The best fitting parameters for the previously collected data [Experiment 1 and data from (3,2) problems reported by Jeffries et al.] are shown in Table 6. For an appropriate test of the model this set of parameters will also be employed for predicting the experimental data, which were collected in the following experiment.

#### **Experiment 2**

Subjects had to perform one of the three different river-crossing tasks: a (3,2), (5,3) or a (4,3) problem. Every legal and illegal move was recorded until the subject reached the goal state of the problem.

#### *Method*

**Subjects.** Another 120 subjects were recruited from introductory psychology classes at the University of Colorado. They received class credit for their participation.

**Table 7.**

	Problem					
Moves	(3,2)		(4,3)		(5,3)	
Observed and predicted means and standard deviations for legal and illegal moves						
Legal						
Mean	16.55	(17.34) <sup>a</sup>	18.90	(18.54) <sup>a</sup>	28.25	(27.37) <sup>a</sup>
SD	6.85	(7.72) <sup>a</sup>	12.48 <sup>a</sup>	(9.51) <sup>a</sup>	17.67	(15.28) <sup>a</sup>
Illegal						
Mean	3.73	(2.86) <sup>a</sup>	3.13	(2.69) <sup>a</sup>	5.63	(5.24) <sup>a</sup>
SD	5.47*	(3.09) <sup>a</sup>	3.60*	(2.77) <sup>a</sup>	6.23	(4.73) <sup>a</sup>
Correlations between observed and predicted state transitions						
Legal	0.98 <sup>b</sup>	(0.97) <sup>a</sup>	0.93 <sup>b</sup>	(0.91) <sup>a</sup>	0.94 <sup>b</sup>	(0.93) <sup>a</sup>
Illegal	0.92 <sup>b</sup>	(0.92) <sup>a</sup>	0.92 <sup>b</sup>	(0.90) <sup>a</sup>	0.92 <sup>b</sup>	(0.91) <sup>a</sup>

\*Significant deviation between observed and predicted statistic (significance level: 0.05)

<sup>a</sup>Predictions computed by the production system model using the general set of parameters shown in Table 6

<sup>b</sup>The best fitting parameters for the particular problem were used

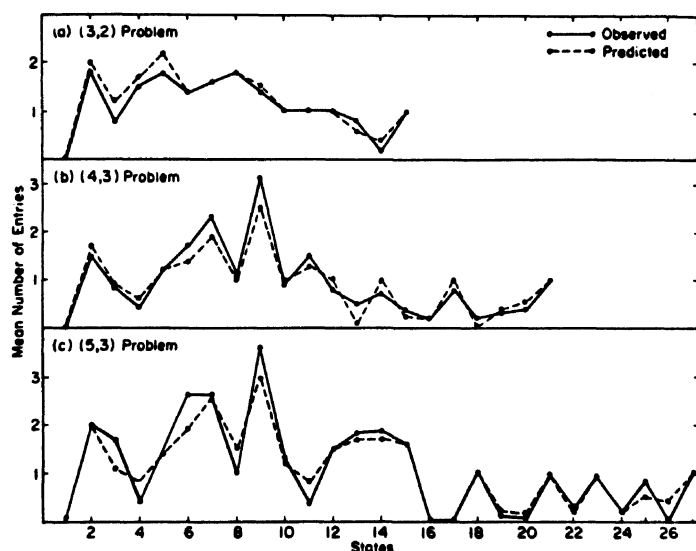


Fig. 5. Mean number of observed and predicted visits to legal states for three different problems

### Procedure

The same apparatus as in Experiment 1 was employed. Each subject was first given general instructions about the experiment. Detailed instructions concerning both the problem and the method of responding were presented on the CRT, which the subject read at his own pace. The instructions for the three experimental groups, which consisted of 40 subjects each, differed only by the number of missionaries ( $n$ ) and cannibals ( $n$ ) and the boat size ( $k$ ) used in the description of the problem. Subjects were presented with a (3,2), (5,3) or a (4,3) problem. In all other respects the statement of the problems was identical.

### Results

For deriving predictions from the simulation model, 250 subjects were simulated for each of the three experimental conditions. The following statistics were employed for comparing observed and predicted group data: Means and standard deviations of the total number of legal moves, means, and standard deviations of the total number of illegal moves, and the average number of visits to the different problem states. In addition, correlations between predicted and observed state transitions were calculated. These statistics were used because they are sufficiently reliable and provide enough empirical constraints for testing the model. Also, other researchers have used the same statistics or a subset of them.

Table 7 shows the observed and the predicted means and standard deviations for legal and illegal moves for the (3,2), (4,3), and the (5,3) problems. The predictions of the model are quite good for all three problems. As can be seen in Fig. 5, the simulation model predicted the mean number of visits to the various problem states quite well for all three problems.

As seen in Table 7, high correlations were found between the observed and the predicted state transitions. The highest correlation of 0.98 was obtained for the (3,2) problem. Even when the illegal moves were included into the calculations and when the set of parameters determined in Experiment 1 was used, none of the correlations dropped below 0.90.

### Discussion

The proposed model formally specifies the knowledge which subjects use for selecting moves. Because river-crossing problems are puzzles which are rather independent of world knowledge, the knowledge is mostly derived from the problem instructions. Two properties are recognized as important characteristics of the problem: The outnumbering restriction and the goal to transfer all travellers to the other side of the river. The balance strategy is a procedural specification of the outnumbering restriction whereas the means-ends driven-move generation is the procedural knowledge obtained from the goal to transfer all travellers to the other bank of the river. Since the untrained subject's understanding of the problem is incomplete and even partially wrong, this procedural knowledge cannot be integrated into a coherent knowledge structure. In certain states the matching productions may contradict one another so that probabilistic resolution rules are employed in the move selection process.

The proposed model assumes similar to Karat's (1983) model that problem solving in transfer tasks is based upon partial understanding which is derived from the procedural encoding of the problem description. In addition to this task specific knowledge, untrained problem solvers use general problem solving skills like means-ends analysis and eliminating move cycles by remembering which states have previously been visited. These assumptions were supported by the correct predictions of the model for three different problems. The irregular structure of the three problems supports the claim that the correct predictions are due to the psychological processing assumptions of the proposed model rather than being an artefact introduced by the particular structure of a given problem.

In comparison to previous simulation models for the river-crossing task (Jeffries et al., 1977; Simon & Reed, 1976), the present model predicts the data at least as well. Several statistics were predicted even better. Contrary to the previous models, which were only successful in predicting subjects' behavior in one river-crossing problem, the present model succeeded in accounting for subjects' moves in three structurally different problems, which had not been used for developing the simulation model. In comparison to flow-chart models, the present production system has several advantages: The model gives an explicit account of the various knowledge components which humans use for solving transformation problems; the model is comparably easy to understand; every production denotes a psychologically meaningful piece of procedural knowledge. The model is decomposable. We may eliminate one or several productions and we would still have an executable simulation program. Thus, predictions can be derived from the reduced production system. By this method we can evaluate the importance of a production as a well specified segment of procedural knowledge with respect to the different river-crossing problems. This was done for the balance production in the move generation process, the level two move evaluation productions, and the long-term memory storage.

As can be seen in Table 8, without the balance production the predictions for the (3,2) problem hardly change. The mean number of moves to the solution state is correctly predicted and the correlation coefficient remains high. However, the predictions for the (4,3) and the (5,3) problem become very inaccurate. In the (4,3) problem the mean number of moves becomes smaller because the highly unbalanced states which must be visited for solving the (4,3) problem are passed more easily. In the (5,3) problem visiting a highly unbalanced state leads to a

**Table 8.** Correlation coefficients and predicted means derived from modified production systems

	(3,2)		(4,3)		(5,3)	
	Mean number of moves to solution	Correlation of state transitions	Mean number of moves to solution	Correlation of state transitions	Mean number of moves to solution	Correlation of state transitions
Observed	16.55	—	18.90	—	28.25	—
Predicted by the complete production system	17.34	0.98	18.54	0.93	27.37	0.94
Predictions without balance production	18.02	0.94	15.74	0.76	32.22	0.78
Predictions without subgoal protection and means-ends evaluation (level 2 Eval.)	26.04	0.74	19.20	0.84	36.78	0.84
Predictions without LTM-storage	23.94	0.87	22.57	0.91	34.19	0.90

dead-end. Therefore, the simulation predicts a higher number of mean number of moves than the simulation model with all productions.

From Table 8 it is also seen that level two move evaluation mostly effects the predictions of the (3,2) and the (5,3) problems. Although the mean remains essentially constant for the (4,3) problem, the correlation drops to .84. Also, the results in Table 8 show that a probabilistic long-term storage of previously visited problem states is equally important in the (3,2), (4,3) and the (5,3) problems. Without long-term storage the mean number of moves increases in all three problems.

### General conclusions

In summary, the proposed production system model is more adequate than the previously developed simulation models for river-crossing tasks: Whereas previous models only predicted subjects' behavior for one problem, the current model yielded excellent fits for three structurally different river-crossing tasks. An equally important advantage of the present model lies in the production system formalism. This formalism permits a decomposition of the model by eliminating one or several productions, so that the significance of various productions for the overall solution path can be assessed. Also, since a similar model (Karat, 1982) has been proposed for the Tower of Hanoi problem, the present process assumptions are supposedly not particular to river-crossing tasks but instead typical for transformation problems. The proposed move generation may also account for the point of view which subjects take in problem solving (Hutchins & Levin, 1981).

The proposed model is expected to be useful for studying learning in river-crossing problems. The model could possibly account for the transfer-effects which are found when subjects have to solve several different problems in a discovery learning task (McDaniel & Schlager, 1985). Because of the production system formalism the learning mechanisms of production de-

signation, strengthening, generalization, discrimination, and composition proposed by Anderson, Kline, and Beasley (1979) could be applied for modeling discovery learning.

*Acknowledgment.* This research was supported by NSF Grant BNS 77-06799 to the second author. The first author was supported by a postgraduate scholarship from the Deutscher Akademischer Austauschdienst. Part of this research has been presented at the meeting of the Society for Artificial Intelligence and the Simulation of Behavior "Production Systems in Psychology" Sheffield/England, 13-15 April, 1981. We would like to thank John Karat, Robin Jeffries, and Mark Schlager for their help and thoughtful comments on this research.

### References

- Anderson, J.R., Kline, P.J., & Beasley, C.M. (1979). A general learning theory and its application to schema abstraction. In Bower, G.G. *The psychology of learning and motivation*. (pp. 277-318). New York: Academic Press.
- Anzai, Y., & Simon, H.A. (1979). The theory of learning by doing. *Psychological Review*, 86, 124-140.
- Atwood, M.E., & Polson, P.G. (1976). A process model for water jug problems. *Cognitive Psychology*, 8, 191-216.
- Atwood, M.E., Masson, M.E.J., & Polson, P.G. (1980). Further explorations with a process model for water jug problems. *Memory & Cognition*, 8 (2), 182-192.
- Duncker, K. (1935). *Zur Psychologie des produktiven Denkens*. Berlin: Springer.
- Ernst, G.W., & Newell, A. (1969). *GPS: A case study in generality and problem solving*. New York: Academic Press.
- Greeno, J.G. (1974). Hobbits and orcs: Acquisition of a sequential concept. *Cognitive Psychology*, 6, 270-292.
- Hutchins, E.L., & Levin, J.A. (1981). Point of view in problem solving. *Proceedings of the third annual conference of the Cognitive Science Society*, (pp. 200-203).
- Jeffries, R., & Karat, J. (1979). A general-purpose program for problem solving experimentation. *Behavior Research Methods and Instrumentation*, 1, 205-207.



- Jeffries, R., Polson, P.G., Razran, L., & Atwood, M.E. (1977). A process model for missionaries-cannibals and other river-crossing problems. *Cognitive Psychology*, 9, 412-440.
- Karat, J. (1983). A model of partial understanding in a problem solving task. *Cognitive Psychology*, 14, 538-559.
- Kieras, D. (1985). The why, when, and how of cognitive simulation: A tutorial. *Behavior Research Methods, Instruments, & Computer*, 17 (2), 279-285.
- McDaniel, M.A., & Schlager, M.S. Discovery learning and transfer of problem solving skills: A cognitive approach. Manuscript University of Notre Dame.
- Newell, A., & Simon, H.A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.
- Polson, P.G., & Jeffries, R. (1982). Problem solving as search and understanding. In Sternberg, R. (Ed.) *Advances in the psychology of human intelligence*. (pp. 367-411) Hillsdale: Lawrence Erlbaum.
- Pratt, T.W. (1975). *Programming languages: Design and Implementation*. Englewood, N.J. Prentice-Hall.
- Schmalhofer, F., Albert, D. Aschenbrenner, K.M., & Gertzen, H. (1986). Process traces of binary choices: evidence for selective and adaptive decision heuristics. *Quarterly Journal of Experimental Psychology*, 36A 59-76.
- Simon, H.A. (1975). The functional equivalence of problem solving skills. *Cognitive Psychology*, 7, 268-288.
- Simon, H.A., & Reed, S.K. (1976). Modeling strategy shifts in a problem solving task. *Cognitive Psychology*, 8, 86-97.
- Weisberg, R.W. (1980). *Memory, Thought and Behavior*. New York. Oxford: Oxford University Press.

Received December 23, 1985/April 6, 1986

## Appendix

### Variables and expressions

<HOBBIT>	:: = <INTEGER>	Number of missionaries and cannibals in problem
<ORC>	:: = <INTEGER>	
<PAIRNUM>	:: = <INTEGER>	
<BOATSIZE>	:: = <INTEGER>	
<BOAT>	:: = RIGHT / LEFT	
<GOAL>	:: = RIGHT / LEFT	
<STATE>	:: = <HOBBIT><ORC><BOAT>	
<DISPLAY>	:: = <STATE>	
<CODE>	:: = <STATE>	
<MOVPROP>	:: = NIL / GEN / APROV	
<MOVE>	:: = <HOBBIT><ORC><MOVPROP>	
<MOVLIST>	:: = NIL / MOVE / MOVE MOVLIST	
<PREMOVE>	:: = <MOVE>	
<LTMSTATE>	:: = NIL / <STATE> / <STATE><LTMSTATE>	
<LGTY>	:: = 0 1 1 2	

### Information in LTM

<BOATSIZE>	characteristics of particular river-crossing problem memory for visited states
<GOAL>	
<PAIRNUM>	
<LTMSTATE>	

### Information in WM

<PREMOVE>	previous move taken list of proposed moves illegal moves after the last legal move
<MOVLIST>	
<LGTY>	

### Information in external & sensory memory

<DISPLAY>	description of display encoding of display
<CODE>	

*Predicates and simple functions (no side effects) occur in Condition-part of productions*

Function name	Verbal description	Arguments	Value returned
EQ NE	pattern match of expressions	ANY ANY	T / F F / T
>	numerical comparison	<INTEGER> <INTEGER>	T / F
SUBGOAL	returns true if all hobbits at goal bank, else F is returned	<STATE>	T / F
LTM	returns true if state is stored in LTM, else F is returned	<STATE>	T / F
PROP	returns nil if MOULIST EQ NIL, else property of first move in list is returned	<MOVLIST>	NIL / GEN / APROV
DIFF	absolute difference between hobbits and orcs	<HOBBIT> <ORC> ANY	INTEGER
LOAD	returns number of hobbits and orcs on bank of boat after critters are loaded into boat	MOVLIST STATE	INTEGER
APPLY	mentally calculate	<MOVLIST> <STATE>	STATE
CRITNUM	returns number of creatures in first move	<MOVLIST>	INTEGER
RANDOM	return T half the time it is called. Pseudo-function	ANY	T / F
FIRST	returns first move of MOVLIST	<MOVLIST>	MOVE

*Functions with side effect – occur only in Action-part of production*

Function name	Verbal description	Arguments	Effect of function
SET PROP	declare first MOVE as generated or approved	<MOVLIST> <MOVPROP>	Set properly first move MOVLIST to MOVPROP
BALANCE		<MOVLIST> <DISPLAY>	Derive best, 2nd best, 3rd best, etc. means-ends move to balanced state and insert into MOVLIST
SAMPLE		<MOVLIST> <CODE>	Describe best, 2nd best, 3rd best means-ends move by sampling hobbits and orcs from state description CODE
REPLACE	modify proposed move	<ANY> <ANY> <MOVLIST>	Argument in Argument and replace by Argument 2.
TAKE		<MOVLIST> <DISPLAY>	Physically execute first MOVE of MOVLIST, if execution successful, set REMOVE to first