Learning numerical progressions*

PAUL C. VITZ and DIANE N. HAZAN

New York University, New York, New York 10003

Learning of simple numerical progressions and compound progressions formed by combining two or three simple progressions is investigated. In two experiments, time to solution was greater for compound vs simple progressions; greater the higher the progression's solution level; and greater if the progression consisted of large vs small numbers. A set of strategies is proposed to account for progression learning based on the assumption S computes differences between integers, differences between differences, etc., in a hierarchical fashion. Two measures of progression difficulty, each a summary of the strategies, are proposed: C_1 is a count of the number of differences needed to solve a progression; C_2 is the same count with higher level differences given more weight. The measures accurately predict in both experiments the mean time to solve 16 different progressions with C_2 being somewhat superior. The measures also predict the learning difficulty of 10 other progressions reported by Bjork (1968).

This paper identifies factors influencing numerical progression learning and presents a simple descriptive model of how this learning takes place. Examples of simple numerical progressions are: (a) 1, 2, 3, 4, 5, ...; (b) 1, 2, 4, 7, 11, 16, ...; (c) 1, 2, 5, 12, 25, ... The solution to Progression a is simply +1 on each trial. Progression b sets up the adjacent differences of 1, 2, 3, 4, 5, ... and adjacent differences between these of +1. Progression c sets up adjacent differences of 1, 3, 7, 13, ...; differences between these of 2, 4, 6, ...; finally, at the next level the differences are +2, ... Very generally, in this paper a simple numerical progression is a sequence of integers which is monotonically increasing or decreasing and which can be solved by taking differences between adjacent integers or between adjacent higher level differences as shown in Table 1. The term "solved" means that at some level the sequence of differences reduces to a constant and by working back to the original sequence it is possible to predict continuously the next integer.

In solving such progressions, the first level of differences is derived from the actual stimulus events called the $d^{(0)}$ level integers, that is, $d_i^{(1)} = d_{i+1}^{(0)} - d_i^{(0)}$. The second level differences are defined: $d_i^{(2)} = d_{i+1}^{(1)} - d_i^{(1)}$, and so on to the solution or k th level. These differences are analogous to derivatives. A first order numerical progression is a sequence of integers such that $d_{i+1}^{(0)} - d_i^{(0)}$ is a constant and is commonly called an arithmetic progression; a second order numerical progression is a sequence such that $d_{i+1}^{(1)} - d_i^{(1)}$ is a constant; etc., for higher orders.

Simple numerical progressions can be combined to form compound progressions. In this paper, a compound double numerical progression is formed by combining two simple progressions so that one progression occurs on all odd trials and the other on even trials. For example, the two simple progressions, NP_1 : 1, 3, 5, 7,

..., and NP_1 : 2, 6, 10, 14, ..., combine to form the compound double progression NP_{1-1} : 1, 2, 3, 6, 5, 10, 7, 14, ..., etc.

Two progressions of the same order can generate integers that differ in size. Consider, NP_{1-A} : 1, 3, 5, 7, ..., and NP_{1-B} : 1, 6, 11, 16, ... The stimulus sequence of Progression NP_{1-A} has zero level integers and first level differences which, when matched against those in NP_{1-B} , are much smaller in size or magnitude. It is possible that if the integers in the stimulus sequence of one progression are larger in magnitude than those in another, this will make it harder to learn, assuming the two progressions are of the same order.

The major significance of numerical progression learning is that conventional stimulus response theories of sequential learning are not able to account for such learning, since no literal stimulus event, no stimulus position, or any group of literal stimuli or stimulus positions ever repeat in such a task. For a more detailed analysis of the difficulties conventional S-R theories have with sequential learning, see Restle (1970). Instead, the S must actually abstract some repeating relationship.

EXPERIMENT I TRIAL BY TRIAL PROCEDURE

Method

Stimuli

The 16 numerical progressions used as stimuli are presented in

Table 1
Solution to a Simple Numerical Progression Obtained by Computing Differences Between Adjacent Integers, Differences Between Adjacent Differences, Etc.

Third Level Differences, $d^{(3)}$	1	1			
Second Level Differences, $d^{(2)}$	1	2	3		
First Level Differences, $d^{(1)}$	1	2	4	7	
First Five Integers	1	2	4	6	15

*This research was supported by Grant NS-07636-04A1 from the National Institute of Neurological Diseases and Stroke, National Institutes of Health. Requests for reprints should be sent to Paul C. Vitz, Department of Psychology, New York University, 4 Washington Place, 9th Floor, New York, New York 10003.

progressions are grouped in pairs which are structurally identical except for the magnitude property.

A three-way analysis of variance, discussed below, was performed on the times to solution of these 16 progressions. The three variables are: simple-compound, small-large, and order: 1, 2, and 3. For each analysis, the 16 progressions in Table 2 were collapsed into a 2 by 2 by 3 design (12 progressions). To fit this design, six of the compound progressions were grouped and treated as two progressions. Specifically, since Nos. 9, 11, and 13 are all compound, small, and of Order 2, each S's mean of these three was treated as a solution for a single pattern; Nos. 10, 12, and 14 are all compound, large, and of Order 2; their mean values were also treated as a single progression score. This procedure permits the 2 by 2 by 3 analysis.

Subjects and Procedure

Twelve undergraduates at New York University, fulfilling a course requirement, served as Ss. The S sat at a desk and was given a booklet of blank notebook size paper. The numbers in the progression were read aloud, one at a time by E; S wrote them in the booklet and, when ready, predicted the next number. This continued until the learning criterion was reached or until 5 min had passed without solution. The S was told all the progressions could be solved and that he would be timed. One practice trial was given on the progression: 1, 2, 3, 4, 5, During the experiment, if an incorrect answer was reported which was close to correct, E reminded S to check his arithmetic. For simple progressions, the learning criterion was two correct predictions; for compound progressions, four correct, two for each simple progression. Using four correct trials, for both simple and compound progressions, is another possible criterion. However, to require four correct trials for learning a simple progression, but to require only two correct for learning the same progression when compounded with another seemed a slight bias against the simple progressions. On the other hand, the present procedure is possibly a slight bias in favor of showing slower learning for compound progressions. Regardless, the effect of a change in criterion would be small, since the average additional time for S to compute an extra two correct predictions for the simple progressions is probably, at most, 10 sec. Even with the present criterion, Ss occasionally gave an extra prediction for the easiest of the simple progressions. Each S received a different counterbalanced order of presentation of the patterns. Learning was timed by starting a clock when S spoke the first number and stopping it when S reached criterion. If S had no hypothesis on a particular trial, he was told to guess. The paper and pencil were provided to avoid requiring S to memorize the stimulus sequence and to avoid overly difficult internal arithmetic.

EXPERIMENT II THE STIMULUS STEM PROCEDURE

This experiment was conducted to replicate Experiment I and to remove certain minor problems created by the trial by trial procedure. In the former procedure, occasionally S would pick up a false set based on a pattern created in the first few events; this could keep S from seeing the correct answer until considerably later than Ss who did not have such a set. The stimulus "stem" procedure reduces these occasional early false sets. The stimulus stem is the first n integers which allow S to solve the progressions with a reasonable degree of certainty. A progression is solved when S has calculated a series of constant differences at any level. We assume two such identical differences constitute a basis for

Table 2
The Numerical Progressions Used in Experiments I and II With the Mean Time and Mean Rank of Solution for Each

	Numerical Progression S = Small, L = Large, 1, 2, 3 = Order	Experiment I*		Experiment II*	
		Mean Time (Sec)	Mean Rank	Mean Time (Sec)	Mean Rank
1 NP ₁ (S):	[1, 3, 5, 7, . . .]	19.3	(7.8)**	8.0	(4.6)**
2 NP ₁ (L):	[2, 5, 8, 11, . . .]	23.6	(7.4)	16.1	(15.4)
3 NP ₂ (S):	[1, 2, 4, 7, 11, . . .]	29.2	(8.0)	28.0	(36.6)
4 NP ₂ (L):	[1, 3, 7, 13, 21, . . .]	50.7	(29.0)	42.2	(72.9)
5 NP ₃ (S):	[1, 2, 4, 8, 15, 26, . . .]	231.3	(86.3)	163.7	(100.6)
6 NP ₃ (L):	[1, 2, 5, 12, 25, 46, . . .]	220.7	(106.0)	144.4	(88.0)
7 NP ₁₋₁ (S):	[1, 2, 5, 6, 9, 10, 13, . . .]	38.2	(17.4)	37.9	(24.0)
8 NP ₁₋₁ (L):	[1, 5, 12, 16, 23, 27, 34, . . .]	59.4	(33.3)	46.6	(23.2)
9 NP ₂₋₁ (S):	[1, 4, 5, 8, 10, 13, 16, 19, 23, . . .]	82.9	(64.8)	70.3	(37.6)
10 NP ₂₋₁ (L):	[1, 5, 6, 10, 13, 17, 22, 26, 33, . . .]	72.0	(31.1)	75.3	(46.6)
11 NP ₂₋₂ (S):	[1, 3, 2, 6, 5, 11, 10, 18, 17, . . .]	60.5	(45.3)	73.6	(75.2)
12 NP ₂₋₂ (L):	[1, 4, 2, 8, 6, 15, 13, 25, 23, . . .]	113.2	(70.5)	115.0	(73.7)
13 NP ₂₋₂ (S):	[1, 3, 4, 8, 10, 16, 19, 27, 31, . . .]	100.7	(45.6)	130.7	(90.4)
14 NP ₂₋₂ (L):	[1, 4, 8, 14, 22, 31, 43, 55, 71, . . .]	240.1	(76.1)	230.6	(82.6)
15 NP ₃₋₂ (S):	[1, 4, 5, 8, 10, 13, 18, 21, 31, 34, 51, . . .]	239.5	(79.2)	249.0	(89.7)
16 NP ₃₋₂ (L):	[1, 5, 6, 10, 12, 16, 22, 26, 39, 43, 66, . . .]	276.0	(59.8)	289.6	(25.5)

*N = 12

**Standard deviations in parentheses

Table 2. Of the six simple progressions in Table 2, two are first order progressions, NP₁; two are second order, NP₂; and two are third order, NP₃. In each case, the two progressions which are of the same order differ in their magnitude properties, the odd numbers being small and the even large. Of the 10 compound progressions, two consist of two simple first order progressions, NP₁₋₁; six consist of two second order progressions, NP₂₋₂; and two of third order progressions, NP₃₋₃. Again, the

reasonable certainty. A longer stem could have been used, but allowing for only two constant differences proved satisfactory. For example, the NP_1 : 1, 3, 5, 7, 9, ... , has a stem denoted by brackets of [1, 3, 5], since two successive identical $d^{(1)}$ values are set up by the first three $d^{(0)}$ integers. The stem of a simple numerical progression of Order k consists of the first $k + 2$ integers presented.

The stem of a compound progression consists of the integers in the stems of the simple progressions out of which it is constructed, e.g., the stem of NP_1 : [1, 5, 9] and NP_1 : [2, 6, 10], when combined to form the stem of a double compound progression, sets up NP_{1-1} : [1, 2, 5, 6, 9, 10]. (No. 7, Table 2.)

Method

The 16 stems in Table 2, each typed on a 5 x 8 in. card, were used as stimuli. Twelve new Ss, New York University undergraduates, were given a card on which was written a progression stem and told to solve the series by predicting the next numbers in the series. They were allowed as much paper as they needed. The timer was stopped when S made a prediction; E then stated if the prediction was correct and the progression solved or if the prediction was wrong. If wrong, the timer was immediately restarted. Thus, the cumulative time worked on the problem was recorded. Otherwise, the procedure was as in Experiment I.

Results

The mean time to solution and the mean rank of solution time for the progression in both experiments are shown in Table 2. The primary result is that time to solution is significantly affected by all three main variables. Specifically, in Experiment I, the mean time to solution for the simple progressions was 95.8 sec and for the compound, 139.9 ($F = 59.35$, $df = 1, 11$, $p < .001$); mean solution time for small progressions was 106.5 and for large 129.2 ($F = 32.83$, $df = 1, 11$, $p < .001$); the mean times for Orders 1, 2, and 3 were 35.1, 76.5, and 241.8, respectively ($F = 120.69$, $df = 2, 22$, $p < .001$). The effect due to order is extreme since the scores of every S increased from order to order by at least 50%. A significant interaction was between simple-compound and order ($F = 6.32$, $df = 2, 22$, $p < .05$) with the large compound progression being relatively more difficult, although the effect was not great.

The results of Experiment II were similar to Experiment I. The mean times for the simple and compound patterns were 67.1 and 142.4 sec ($F = 65.4$, $df = 1, 11$, $p < .001$); the mean for the small progressions was 96.4 vs 113.2 for large ($F = 7.59$, $df = 1, 11$, $p < .05$). The means for Orders 1, 2, and 3 were 27.1, 75.4, and 211.8, respectively ($F = 98.38$, $df = 2, 22$, $p < .01$); and simple-compound interacted with small-large ($F = 6.92$, $df = 1, 11$, $p < .05$). In Experiment II, order did not interact with small-large.

Discussion

The effects due to solution level and to compounding

are not especially surprising, since higher level solutions require more computation, and a compound containing two progressions should be harder to learn than just one progression. More interesting is the fact that large numbers make a progression more difficult. Perhaps, the large numbers had more digits and hence required more computation. To some extent this is possible, since the large and small progressions were not quite identical in this respect. But, it is also possible that large numerical differences are intrinsically harder to work with, e.g., Groen and Parkman, 1972, show that adding two digits is affected by digit size. (See also Moyer & Landauer, 1967; Sekuler, Rubin, & Armstrong, 1971.)

The interactions are not always interpretable, since the more difficult patterns often were not solved by the cutoff time of 300 sec. This ceiling effect was fairly strong, since in the three or four most difficult patterns, it was common for 40%-50% of the Ss to fail to solve the pattern in under 5 min. Disregarding the ceiling effect, in general, it was the case that, as the progressions became compound, large, and higher order, they became relatively more difficult.

THE MODEL

The strategies provide a specific solution for simple progressions and for the compound progressions of the type treated here.

Strategy 1

This is the obvious strategy of discrete differencing of adjacent differences shown in Fig. 1, which is successful with simple progressions but fails with compound progressions. For example, NP_7 : 1, 2, 5, 6, 9 has $d^{(1)}$ elements of 1, 3, 1, 3, ... ; $d^{(2)}$ elements of 2, -2, 2, -2, ... ; $d^{(3)}$ elements of -4, 4, -4, 4, ... , etc. The differences get larger since two or more compounded progressions create a nonmonotonic series at some level, often as early as $d^{(1)}$. This break in monotonicity is the cue for the next strategy.

Strategy 2A

When the particular $d^{(i)}$ is computed, which signals the violation of monotonicity, $d^{(i)}$ elements are compared which are g elements apart. These jump comparisons are made retroactive to the first element in the series; they are made for jumps of Size 1 and 2, i.e., the stop rule for the size of g is 2. If any such jump comparisons result in differences of Size 0, that part of the compounded progression has been solved. Again, consider NP_7 (NP_8 has the same structure). After the asterisk Strategy 2A is started, which compares every other $d^{(1)}$ retroactive to the first in the sequence, and this results in solution for one half of the progression and then the other. The solution is +1, +3, +1, +3, etc. If the two compounded progressions had been learned as separate simple progressions, the solution would have been +4 for each. Thus, because compounded

Table 3
Solution of Numerical Progression 13 [1,3,4,8,10,16,19]

Level	Strategies 1, 2A, 2B	Level	Strategy 2C
d(2)	-1	2	1
d(1)	2 1 4* 2**	2 1 4 2 6 3	
d(0)	1, 3, 4, 8, 10, 16***	1, 3, 4, 8, 10, 16, 19	

*Strategy 1: violation of monotonicity.

**Strategy 2A: no identical element sequence at this level.

***Strategy 2B: no constant jump difference sequence at lower level. Analysis returns to d(1) and computes, via Strategy 2C, the next level of jump differences at d(2), which leads to solution.

progressions interact with each other, the "compound" solution is not the same as the solution for the same progressions considered in isolation. If only one set of jump d(1) values results in immediate solution, then the remaining set of differences is differenced by Strategy 1 at higher levels until solved. Progressions 9, 10, 15, and 16 are of this type.

Strategy 2B

If no set of d(i) jump comparisons consists of identical values, the event which signals this failure is identified with a double asterisk, **, and the jump difference search moves to Level d(i-1), i.e., the immediately preceding difference level. If both series of these lower order jump differences are constant, solution has been reached. An example is Progression 14. If only one series is constant, one progression is solved and the remaining series is differenced to solution. (Or until a violation of monotonicity requires application of Strategy 2A again.) The reason for introducing the lower level jump differences is that if a solution can be obtained lower in the hierarchy it is simpler, i.e., can be done with fewer computations.

Strategy 2C

If Strategy 2B fails, noted by ***, then the search returns to Level d(i) and evaluation of the next level of jump differences, d(i+1), is made. The solution of Progression 13 in Table 3 is an example of this strategy. (Progressions 11 and 12 are also of this type.)

In applying the strategies, occasionally differences are needed at a level lower than that at which the strategy is operating. It is assumed that these necessary differences are computed first. All the progressions treated here are solved by Strategies 1 and 2, but in the case of more complex progressions where 2C fails, then Strategies 2A, B, and C can be applied at still higher levels. The process should stop before solution in cases where there is no solution or when only a very high order solution exists. The arbitrary stop rule is that no d(i) is computed for j > 3.

Measures of Progression Difficulty: C₁ and C₂

Two simple summaries of the preceding strategies are proposed as indices of a progression's learning difficulty or complexity. The first is a simple count of the differences that result from application of the strategies with each difference involving a two-digit number receiving a weight of two. This count, C₁, for each progression is shown in Table 4. Differences set up by the strategies but not actually involved in the final solution hierarchy also count. For example, Progression 7, which has the d(1) values of 1,3,1*, also has a d(2) value of +2 just prior to the violation of monotonicity. This one extra difference occurs in all the compound progressions; Progression 14 also has three other d(2) differences created in this manner. The measure C₁ increases with solution level with the number of compounded progressions, and, because of double weight for differences involving two-digit numbers, it often is slightly greater for large as compared to small progressions. However, it does not always differentiate large and small progressions, e.g., Progressions 1 and 2 and 11 and 12 have the same C₁ score but differ in magnitude and in solution time. The measure C₁ correlates .836 and .923 with mean time to solution in Experiments I and II. C₁ correlates .888 and .936 with mean rank of solution time in the same two experiments.

The measure C₁ is based on the assumption that the difficulty of computing a difference is unaffected by its level, i.e., a d(1) and a d(3) of the same size count equally. But higher differences presumably are more difficult to compute, perhaps because they are harder to

Table 4
Summary of the C₁ and C₂ Measures of Progression Difficulty (Complexity)

Progression Number	Progression Difficulty Measure							
	Level			C ₁ Total	Level			C ₂ Total
	1	2	3		1	2	3	
1	2	—	—	2	6	—	—	6
2	2	—	—	2	6	—	—	6
3	3	2	—	5	9	14	—	23
4	4	2	—	6	12	14	—	26
5	5	3	2	10	15	21	30	66
6	6	4	2	12	18	28	30	76
7	4	1	—	5	12	7	—	19
8	7	1	—	8	21	7	—	28
9	9	3	—	12	27	21	—	48
10	10	3	—	13	30	21	—	51
11	6	3	—	9	18	21	—	39
12	6	3	—	9	18	21	—	39
13	9	5	—	14	27	35	—	62
14	10	9	1	20	30	63	15	108
15	13	5	2	20	39	35	30	104
16	14	5	2	21	42	35	30	107

Note—Progression numbers are from Table 2.

perceive or notice, or because sequences of lower level differences do not seem very patterned when the solution is at a high level, which tends to divert attention to other strategies, or because working back to the original sequence to make a prediction gets harder the higher the level from which working back begins. Evidence for this is that Progressions 5 and 6 are much more difficult than Progressions 9, 10, 11, 12, and 13, but all of these have about the same C_1 score. Thus, a modification of C_1 is proposed which gives extra weight to higher differences. This second measure, C_2 , is based on the assumption that every C_1 difference is weighted (multiplied) by the number of numbers below it plus one. This assumes difficulty of computing a difference is adequately estimated by the number of preceding lower level differences plus one for the difference itself. Thus, $d^{(1)}$ is weighted by $2 + 1$ or 3 ; $d^{(2)}$ is weighted by $3 + 3 + 1$ or 7 ; $d^{(3)}$ by 15 , etc. The C_2 scores for each progression are also shown in Table 4. The major difference is that C_2 is proportionally greater for progressions with a high level solution, especially $d^{(3)}$ level progressions. The measure C_2 correlates .937 and .967 with mean solution time in Experiments I and II; it correlates .939 and .957 with the mean ranks. All four of these C_2 correlations are higher than the corresponding C_1 correlations and result in an average improvement in variance of about 10%. (The four correlations between C_2 and the data, with C_1 partialled out, are all highly significant. The t values for the partial correlations range from 8.45 to 2.50, p from $< .001$ to $< .05$, two-tailed, $df = 13$.) Progression 5 is still moderately discrepant from its predicted score apparently because it suggested a simple geometric progression as the answer. That is, several Ss reported 16 as the fifth event and had some difficulty breaking this set. This effect was not as strong in Experiment II, since the stem tends to contradict the geometric hypothesis from the start.

BJORK'S EXPERIMENT

Bjork (1968) presented 12 progressions to 59 Ss in a trial by trial procedure. Of these 12, 2 are not numerical progressions of the type treated here; they are excluded (i.e., Bjork's Progressions 10 and 12 from his Fig. 1 are cyclic patterns. Actually, the strategies do solve these and including them does not hurt the predictions, but to do so implies the strategies are adequate for cyclic patterns in general. We have not investigated this possibility systematically and have no data to test it, hence the exclusion.) The apparatus for presenting the sequence was a 12-in. circle mounted on a board. The digits 0, 1, . . . , 9 were printed at equal intervals around the circle and just inside it. Next to each digit was a light. A digit was designated by turning on its adjacent light. When the switch that activated the lights was turned from one number to the next, all numbers in between flashed momentarily, enabling S to observe the

direction of change. The nature of this apparatus is important, since it meant that any particular stimulus number, N , appeared as N (modulo 10). Hence, if two successive numbers of a sequence were 8 and 12, the Ss saw 8 and 2. For this reason, we assume, as does Bjork, that S began learning by computing the successive differences between the numbers and the model is applied to these differences. Although some of Bjork's progressions are triple compound and require a jump difference of two, the hierarchical difference analysis provided by the strategies are straightforward and a summary of their predicted processing complexity is given in Table 5. Progressions 1 and 4 are simple and of Order 2 and 1, respectively. The others are double or triple compound progressions of Order 1 or 2 or a mixture of both orders. Their solutions are usually of a type similar to Progressions 7 and 8 in Table 2, i.e., $(+k_1, +k_2)$, etc.; $(+k_1, -k_2)$, etc. To test the predicted difficulty values, the difficulty of learning each sequence was summarized from Bjork's Fig. 1. His figure presents the mean proportion of errors from Trial 1 to Trial 14, by which trial the curves were essentially asymptotic. The mean proportion of errors over these trials shown in Table 5 is used as a measure of the pattern's difficulty. The correlation between C_1 and this proportion is .890, while C_2 correlates .895. (These correlations are not significantly different.) Other qualitative support for the strategies is that the subrules identified by Bjork as the starting point of his analysis are also those that the strategies identify.

EXPERIMENT III

This experiment is another test of the model on the 10 progressions used by Bjork.

Method

A sequence from each of the 10 progressions shown in Table 5 was typed on a 3 x 5 in. card. The actual stimulus sequence consisted of the first six $d^{(1)}$ integers, since it is assumed that it is these with which Bjork's Ss began their learning. Each number was preceded by the appropriate + or - sign. The new Ss ($N = 12$) were shown the 10 cards in a randomized order, and they ranked the number strings from the simplest to the most complex, using whatever criteria for "simple" and "complex" they wished.

Results

The mean complexity rank for each progression is shown in Table 5. The correlations of C_1 and C_2 with mean complexity are .949 and .921, respectively. (The partial correlation between C_1 and complexity with C_2 partialled out is significant, i.e., $t = 3.75$, $p < .02$, two-tailed, $df = 13$.) The only noticeably deviant progression is the first: 1.2.3.4. . . . , which has a judged complexity lower than predicted, perhaps due to its extreme familiarity. The correlations between the mean proportion of errors derived from Bjork's study and the

Table 5
**Progressions from Bjork (1968) With Their $d(1)$ Level Integers; Predicted Difficulty, C_1 and C_2 ;
 Bjork's Mean Proportion of Errors; Judged Complexity, Mean Rank, from Experiment III**

Numerical Progressions from Bjork's Fig. 1	$d(1)$: Where Analysis Begins	Predicted Difficulty		Mean of Average Proportion of Errors Trials 1-14 Bjork's Fig. 1		Judged Complexity Mean Rank Experiment III	
		C_1	C_2				
1 [1,2,4,7,] ...	1,2,3	5	23	.153	(.26)*	2.67	(2.15)*
2 [0,2,1,4,2,6,3,] ...	2,-1,3,-2,4,-3	10	53	.354	(.32)	8.25	(1.06)
3 [1,5,7,11,13,] ...	4,2,4,2	4	19	.289	(.36)	3.58	(0.67)
4 [1,4,7,] ...	3,3	2	6	.122	(.28)	1.17	(0.39)
5 [0,3,2,3,6,5,6,] ...	3,-1,1,3,-1,1	6	25	.284	(.38)	5.92	(1.38)
6 [0,4,2,1,5,3,2,] ...	4,-2,-1,4,-2,-1	6	25	.241	(.37)	6.25	(1.14)
7 [2,7,6,11,9,14,11,] ...	5,-1,5,-2,5,-3	8	39	.393	(.33)	7.00	(1.54)
8 [1,3,2,6,3,9,4,] ...	2,-1,4,-3,6,-5	10	53	.530	(.30)	9.58	(0.79)
9 [1,2,4,5,7,] ...	1,2,1,2	4	19	.190	(.33)	3.00	(0.60)
10 [0,2,3,6,8,12,15,] ...	2,1,3,2,4,3	10	53	.480	(.28)	7.58	(0.67)

*Standard deviations are in parentheses.

mean complexity is .901. This suggests that large differences in procedure cannot override the stimulus properties of the progression's structure. Progressions 3 and 9 are structurally identical but Progression 3 with slightly larger numbers is slightly harder. Thus, there is again modest evidence for a digit size effect.

Discussion

In Bjork's experiment, C_1 and C_2 did not differ significantly in their ability to predict progression difficulty, but in Experiment III, C_1 was slightly superior. We attribute this to the fact that Bjork's progressions hardly vary in their level of solution. For example, none of his progressions have a solution as high as the third level. Also, all but one of his progressions required the evaluation of at least one $d(2)$ difference. Still, this result is surprising and qualifies the previous case for the C_2 measure.

The strategies contain two places where stop rules are applied: one to stop the testing for jump differences at Size 2 and the other to stop hierarchical differing at the third level. These values are ad hoc and, however necessary and reasonable, they are not directly tested nor do they have any theoretical basis. Thus, a rationale for a general stop rule will be proposed. We assume that the major factor that leads S to stop an ongoing strategy, besides discovery of solution, is the complexity or cognitive load associated with the strategy. Presumably, when this crosses some maximum value, S cannot or will not continue. The two complexity measures are equally applicable on each trial prior to solution and their values serve as a continuously present index of cognitive load; a fixed C_1 or C_2 value, plus or minus some fluctuation, could serve as a constant basis for a stop rule.

RELATED APPROACHES

The present model has major similarities to the hierarchical coding system of Simon and Kotovsky

(1963), Simon and Sumner (1968), Vitz and Todd (1969), and Restle (1970). [See Simon (1972) for a discussion of the similarities of all these systems.] Because numerical progressions involve metric stimuli instead of letters or other nominal events, the present model is closest to Simon and Sumner and to Restle, and it can be considered a variant of their systems. Although neither Simon or Restle treat numerical progressions directly, hierarchical differing is a property of both, and presumably each system could be extended to such sequences. The present study differs from the preceding theorists by suggesting a weighting of the different levels by proposing rather specific strategies for solving compound numerical progressions and by providing some limited evidence for a digit size effect.

REFERENCES

- Bjork, R. A. All-or-none subprocesses in the learning of complex sequences. *Journal of Mathematical Psychology*, 1968, 5, 182-195.
- Groen, G. J., & Parkman, J. A chronometric analysis of simple addition. *Psychological Review*, 1972, 79, 329-343.
- Moyer, R. S., & Landauer, T. K. Time required for judgments of numerical inequality. *Nature*, 1967, 215, 1519-1520.
- Restle, F. Theory of serial pattern learning: Structural trees. *Psychological Review*, 1970, 77, 481-495.
- Sekuler, R., Rubin, E., & Armstrong, R. Processing numerical information: A choice time analysis. *Journal of Experimental Psychology*, 1971, 90, 75-80.
- Simon, H. A. Complexity analyses and the representation of patterned sequences of symbols. *Psychological Review*, 1972, 79, 369-382.
- Simon, H. A., & Kotovsky, K. Human acquisition of concepts for sequential patterns. *Psychological Review*, 1963, 70, 534-546.
- Simon, H. A., & Sumner, R. K. Pattern in music. In B. Kleinmuntz (Ed.), *Formal representation of human judgment*. New York: Wiley, 1968.
- Vitz, P. C., & Todd, T. C. A coded element model of the perceptual processing of sequential stimuli. *Psychological Review*, 1969, 76, 433-449.

(Received for publication October 25, 1972;
 revision received July 6, 1973.)