MODELS FOR OPTIMIZING THE LEARNING PROCESS

by

G. J. Groen and R. C. Atkinson

TECHNICAL REPORT NO. 92

February 11, 1966

PSYCHOLOGY SERIES

INSTITUTE FOR MATHEMATICAL STUDIES IN THE SOCIAL SCIENCES
STANFORD UNIVERSITY
STANFORD, CALIFORNIA

Models for Optimizing the Learning Process[1]

G. J. Groen and R. C. Atkinson

Stanford University

Abstract

This paper is concerned with showing how certain instructional problems can be reformulated as problems in the mathematical theory of optimization. A common instructional paradigm is outlined and a notational system is proposed which allows the paradigm to be restated as a multi-stage decision process with an explicit mathematical learning model embedded within it. The notion of an optimal stimulus presentation strategy is introduced and some problems involved in determining such a strategy are discussed. A brief description of dynamic programming is used to illustrate how optimal strategies might be discovered in practical situations.

Although the experimental work in the field of programmed instruction has been quite extensive, it has not yielded much in the way of unequivocal results. For example, Silberman (1962), in a summary of 80 studies dealing with experimental manipulations of instructional programs, found that 48 failed to obtain a significant difference among treatment comparisons. When significant differences were obtained, they seldom agreed with findings of other studies on the same problem. The equivocal nature of these results is symptomatic of a deeper problem that exists not only in the field of programmed instruction but in other areas of educational research.

An instructional program is usually devised in the hope that it optimizes learning according to some suitable criterion. However, in the absence of a well-defined theory, grave difficulties exist in interpreting the results of experiments designed to evaluate the program. Usually the only hypothesis tested is that the program is better than programs with different characteristics. In the absence of a theoretical notion of why the program tested should be optimal, it is almost impossible to formulate alternative hypotheses in the face of inconclusive or contradictory results. Another consequence of an atheoretical approach is that it is difficult to predict the magnitude of the difference between two experimental treatments. If the difference is small then it may not turn out to be significant when a statistical test is applied. However, as Lumsdaine (1963) has pointed out, lack of significance is often interpreted as negative evidence.

What appears to be missing, then, is a theory that will predict the conditions under which an instructional procedure optimizes learning. A theory of this type has recently come to be called a *theory of instruction*.

It has been pointed out by several authors (e.g., articles by Gage, 1963 and Hilgard, 1964), that one of the chief problems in educational research has been a lack of theories of instruction. For example, Bruner (1964) has characterized a theory of instruction as a theory that sets forth rules concerning the most efficient way of achieving knowledge or skill; these rules should be derivable from a more general view of learning. However, Bruner makes a sharp distinction between a theory of learning and a theory of instruction. A theory of learning is concerned with describing learning. A theory of instruction is concerned with prescribing how learning can be improved. Among other things, it prescribes the most effective sequence in which to present the materials to be learned and the nature and pacing of reinforcement.

While the notion of a theory of instruction is relatively new, optimization problems exist in many other areas and have been extensively studied in a mathematical fashion. Within psychology, the most prominent example is the work of Cronbach and Gleser (1964) on the problem of optimal test selection for personnel decisions. Outside psychology, optimization problems occupy an important place in areas as diverse as physics, electrical engineering, economics and operations research. Despite the fact that the specific problems vary widely from one field to another, several mathematical techniques have been developed that can be applied to a broad variety of optimization problems.

The purpose of this paper is to indicate how one of these techniques, dynamic programming, can be utilized in the development of theories of instruction. Dynamic programming was developed by Bellman and his associates (Bellman, 1957, 1961; Bellman and Dreyfus, 1962) for the solution of a class of problems called multi-stage decision processes. Broadly speaking, these are processes in which decisions are made sequentially, and decisions made early in the process affect decisions made subsequently.

We will begin by formalizing the notion of a theory of instruction and indicating how it can be viewed as a multi-stage decision process. This formalization will allow us to give a precise definition of the optimization problem that arises. We will then consider in a general fashion how dynamic programming techniques can be used to solve this problem. Although we will use some specific optimization models as illustrations, our main aim will be to outline some of the obstacles that stand in the way of the development of a quantitative theory of instruction and indicate how they might be overcome.

Multi-Stage Instructional Models

The type of multi-stage process of greatest relevance to the purposes of this paper is the so-called discrete N-stage process. This process is concerned with the behavior of a system that can be characterized at any given time as being in state w. This state may be univariate but is more generally multivariate and hence is often called a state vector (the two terms will be used interchangeably). The state of this system is determined by a set of decisions. In particular, every time a decision d (which may also be multivariate) is made, the state of the system is transformed. The new state is determined by both d and w and will be denoted by $T(w,d)$. The process consists of N successive stages. At each of the first $N-1$ stages, a decision d is made. The last stage is a terminal stage in which no decision is made. The process can be viewed as proceeding in the following fashion: Assume that, at the beginning of the first stage the system is in state w_1. An initial decision d_1 is made. The result is a new state w_2 given by the relation:

$$w_2 = T(w_2,d_1) .$$

We are now in the second stage of the process, so a second decision d_2 is made resulting in a new state w_3 determined by the relation

$$w_3 = T(w_2,d_2) .$$

The process continues in this way until finally:

$$w_N = T(w_{N-1},d_{N-1}) .$$

If each choice of d determines a unique new state, $T(w,d)$, then the process is deterministic. It is possible, however, that the new state is probabilistically related to the previous state. In this nondeterministic case, it is also necessary to specify for each stage i a probability distribution $Pr(w_i|w_{i-1},d_{i-1})$.

In a deterministic process, each sequence of decisions d_1, d_2, ... d_{N-1} and states w_1, w_2, ..., w_N has associated with it a function that has been termed the criterion or return function. This function can be viewed as defining the utility of the sequence of decisions. The optimization problem one must solve is to find a sequence of decisions that maximizes their criterion function. The optimization problem for a nondeterministic process is similar except that the return function is some suitable type of expectation.

In order to indicate how an instructional process can be considered as an N-stage decision process, a fairly general instructional paradigm will be introduced. This paradigm is based on the type of process that is encountered in computer-based instruction. In instruction of this type, a computer is programmed to decide what will be presented to the student next. The decision procedure, which is given in diagrammatic form in Fig. 1, is based on the past stimulus-response history of the student. It should be noted that this paradigm contains, as special cases, all other programmed instructional techniques currently in vogue. It may also correspond to the behavior of a teacher making use of a well-defined decision procedure.

It will be assumed that the objective of the instructional procedure is to teach a set of concepts, and that the instructional system has available to it a set of stimulus materials regarding these concepts. For brevity of expression, a concept will be said to be presented whenever material relevant to the concept is presented. We can thus view the presentation of materials available to the system as a set of concepts S.

We will define a stage of the process as being initiated when a decision is made regarding which concept is to be presented and terminated when the history file is updated with the outcome of the decision. In order to completely define the instructional system we need to define:

1. The set S of all possible stimulus presentations.
2. The set A of all possible responses that can be made by the student.
3. The set H of histories. An element of H need not be a complete history of the student's behavior. It may only be a summary. In the extreme case of a linear program it only contains a record

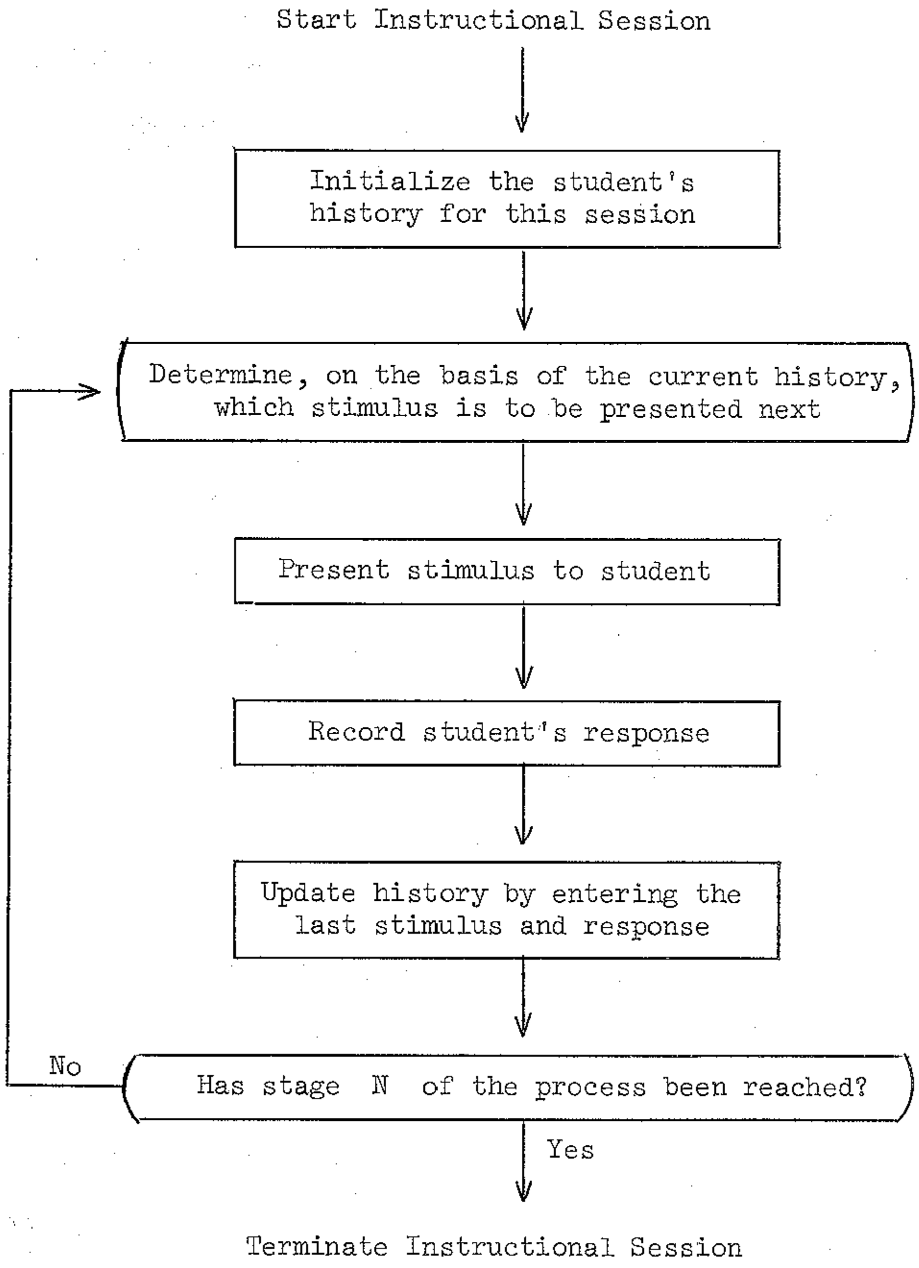
Start Instructional Session
Initialize the student's history for this session
Determine, on the basis of the current history, which stimulus is to be presented next
Present stimulus to student
Record student's response
Update history by entering the last stimulus and response
No
Has stage N of the process been reached?
Yes
Terminate Instructional Session

Figure 1. Flow diagram for an instructional system.

the stage the process is in.

4. A function δ of H onto S. This defines the decision procedure used by the system to determine the stimulus presentation on the basis of the history.

5. A function μ of $S \times A \times H$ onto H. Thus function updates the history.

Thus, at the beginning of stage i the history can be viewed as being in stage h_i. A decision is then made to present $s_i = \delta(h_i)$, a response a_i is made to s_i and the state of the system is updated to $h_{i+1} = \mu(s_i, a_i, h_i)$.

In a system such as this, the stimulus set S is generally predetermined by the objectives of one's instructional procedure. For example, if the objective is to teach a foreign language vocabulary, then S might consist of a set of words from the language. The response set A is, to a great extent, similarly predetermined. Although there may be some choice regarding the actual response mode utilized (e.g., multiple choice versus constructed response), this problem will not be considered here. The objectives of the instructional procedure also determine some criterion of optimality. For example, in our vocabulary example this might be the student's performance on a test given at the end of a learning session. The optimization problem that will be the main concern of this paper is to find a suitable decision procedure for deciding which stimulus s_i to present at each stage of the process, given that S, A and the optimality criterion are specified in advance. Such a decision procedure is called a strategy. It is determined by the set of possible histories, H, the decision function δ, and the updating function μ.

For a particular student, the stimulus presented at a given stage and the response the student makes to that stimulus can be viewed as the observable outcome of that stage of the process. For an N-stage process, the sequence $(s_1, a_1, s_2, a_2, \cdots, s_{N-1}, a_{N-1})$ of outcomes at each stage can be viewed as the outcome of the process. The set of all possible outcomes of an instructional procedure can be represented as a tree with branch points occurring at each stage for each possible stimulus presentation and each possible response to a stimulus. An example of such a tree is given in Fig. 2 for the first two stages of a process with stimulus presentations, s, s' and two responses a, a'.

The most complete history would contain, at the beginning of each stage, a complete account of the outcome of the procedure up to that stage. Thus, h_i would consist of some sequence $(s_1, a_1, s_2, a_2, \cdots, s_{i-1}, a_{i-1})$. Ideally, one could then construct a decision function δ which specified, for each possible outcome, the appropriate stimulus presentation s_i. However, two problems emerge. The first is that the number of outcomes increases rapidly as a function of the number of stages. For example, at the 10th stage a process such as that outlined in Fig. 2, we would have 4^{10} outcomes. The specification of a unique decision for each outcome would clearly be a prohibitively lengthy procedure. As a result, any practical procedure must classify the possible outcomes in such a way as to reduce the size of the history space. Apart from the problem of the large number of possible outcomes, one is also faced with the problem that many procedures do not store as much information as others. For example, in a linear program in which all studens are run in lockstep, it

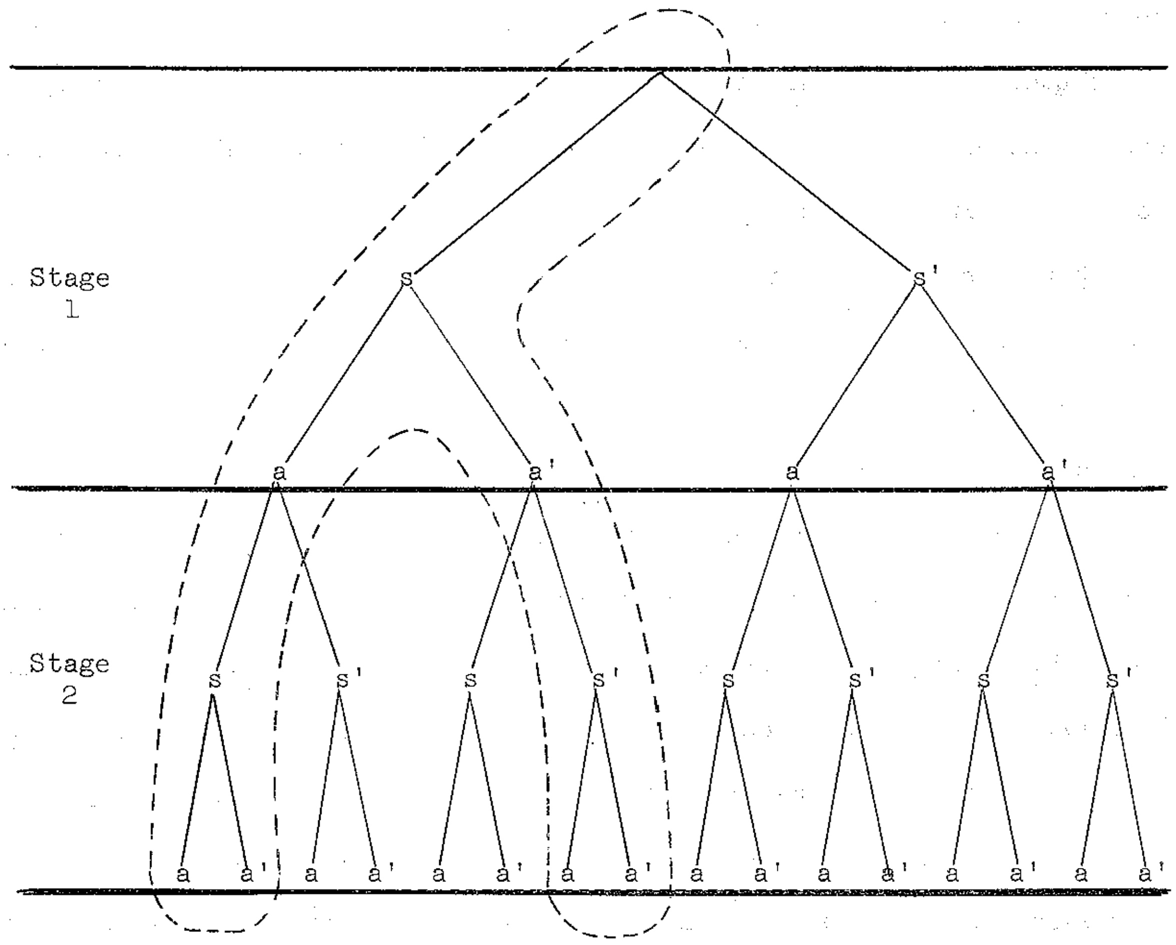

Figure 2. Tree diagram for the first two stages of a process with two stimuli s and s', and two responses a and a'. The dotted lines enclose the subtree generated by a possible response-sensitive strategy.

is not possible to make use of information regarding the student's responses. In general, instructional systems may be classified into two types: those that make use of the student's response history in their stage-by-stage decisions and those that do not. The resulting strategies may be termed _response insensitive_ and _response sensitive_. A response insensitive strategy can be specified by a sequence of stimulus presentations $(s_1, s_2, \cdots, s_{N-1})$. A response sensitive strategy can be represented by a subtree of the tree of possible outcomes. An example is given in Fig. 2. There are two chief reasons for making this distinction. The first is that response insensitive strategies are less complicated to derive. The second is that response insensitive strategies can be completely specified in advance and so do not require a system capable of branching during an actual instructional session.

While this broad classification will be useful in the ensuing discussion, it is important to note that several types of history space are possible within each class. Even if the physical constraints of the system are such that only response insensitive strategies can be considered, it is possible to define many ways of "counting" stimulus presentations. The most obvious way is to define the history at stage i as the number of times each stimulus has been presented. A more complicated procedure (which might be important in cases where stimuli were easily forgotten) would be to also count for each stimulus the number of other items that had been presented since its most recent presentation.

The discussion up to this point has been concerned mainly with the canonical representation of an instructional system and a deliberate effort has been made to avoid theoretical assumptions. While this leads to some insight into the nature of the optimization problem involved, the multi-stage process cannot be sufficiently well defined to yield a solution to the optimization problem without imposing theoretical assumptions. It will be recalled that, in order to define a multi-stage process, it is necessary to specify a transformation $w_{i+1} = T(w_i,d_i)$ given the state and decision at stage i. In order to optimize the process it is also necessary to be able to state at each stage the effect of a decision upon the criterion function. The purpose of most instructional systems is to maximize each student's performance on some aptitude that can be operationally defined in terms of a test that is administered at the end of the procedure. As a result, the simplest criterion function to use is one which depends only on the final state of the system. This function will be called the <u>terminal</u> <u>return</u> <u>function</u> and denoted $\varphi(w_N)$. While this can be stated as a function of the final state, this final state is dependent upon the sequence of decisions that have gone before. However, any two outcomes that result in the same final state yield identical values of $\varphi(w_N)$.

The main purpose of the transformation T is to provide a means of predicting the final state. If T is deterministic then the sequence of optimum decisions could be determined by enumerating in a tree diagram all possible outcomes and computing $\varphi(w_N)$ for each path. A path that maximized $\varphi(w_N)$ would yield a sequence of decisions corresponding to appropriate nodes of the tree. If T were nondeterministic then a strategy σ would yield a subtree similar to that for a response-sensitive strategy. Each subtree would define a probability distribution over the w and thus an expected terminal return:

$$E(\varphi(w_N)|\sigma) = \sum_{w_N \in W} \varphi(w_N)P(w_N|\sigma) \quad , \tag{1}$$

could be computed for each strategy σ. In either case, this process of simple enumeration of the possible branches of a tree is impossible in any practical situation since too many alternative paths exist, even for N reasonably small. The problem of developing a feasible computational procedure will be discussed in the next section. The problem of immediate concern is the most satisfactory way of defining the state space w and the transformations $T(d_i,w_i)$.

At first sight, it would seem that w_i could be defined as the history at stage i and $T(w_i,d_i)$ as the history updating rule. However, while this might be feasible in cases where the history space is either specified in advance or subject to major constraints, it has the severe disadvantage that it necessitates an ad hoc choice of histories and, without the addition of theoretical assumptions, it is impossible to compare the effectiveness of different histories. Even if the history space is predetermined, such as might be the case in a simple linear program where all a history can do is "count" the occurrences of each stimulus item, it is necessary to make some theoretical assumption regarding the precise form of $\varphi(w_N)$.

One way to avoid problems such as this is to introduce theoretical assumptions regarding the learning process explicitly in the form of a mathematical model. In the context of an N-stage process a learning model consists of: 1) a set of learning states Y; 2) a usually nondeterministic response rule which gives (for each state of learning) the probability of a correct response to a given stimulus; and 3) an updating rule which provides a means of determining the new learning state (or distribution of states) that results from the presentation of a stimulus, the response the student makes, and the reinforcement he receives. At the beginning of stage i of the process, the student's state is denoted by y_i. After stimulus s_i is presented, the student makes a response a_i the probability of which is determined by s_i and y_i. The learning state of the student then changes to $y_{i+1} = T(y_i, s_i)$. Models of this type have been found to provide satisfactory descriptions for a variety of learning phenomena in areas such as paired-associate learning and concept formation. Detailed accounts of these models and their fit to empirical phenomena are to be found in Atkinson, Bower, and Crothers (1965), Atkinson and Estes (1963), and Sternberg (1963).

In the example of the learning of a list of vocabulary items, the two simplest models that might provide an adequate description of the student's learning process are the single-operator linear model (Bush and Sternberg, 1959) and the one-element model (Bower, 1961; Estes, 1960). In the single-operator linear model, the set Y is the closed unit interval $[0, 1]$. The states are values of response probabilities. Although these probabilities can be estimated from group data, they are unobservable when individual subjects are considered. If, for a particular stimulus j, $q_i^{(j)}$ is the probability of an error at the start of stage i and that item is presented, then the new state (i.e., the response probability) is given by

$$q_{i+1}^{(j)} = \alpha q_i^{(j)} \qquad (0 < \alpha \leq 1). \quad (2)$$

If $q_1^{(j)}$ is the error probability at the beginning of the first stage, then it is easily shown that

$$q_i^{(j)} = q_1^{(j)} \alpha^{n_i^{(j)}} \qquad (3)$$

where $n_i^{(j)}$ is the number of times the item has been presented and reinforced prior to stage i. The response rule is simply that if a subject is in state q_i with respect to a stimulus and that stimulus is presented then he makes an error with probability q_i. This model has two important properties. The first is that, although the response rule is nondeterministic, the state transformation that occurs as the result of an item presentation is deterministic. The second is that the model is response insensitive since the same transformation is applied to the state whether a correct or incorrect response occurs. The only information that can be used to predict the state is the number of times an item has been presented.

In the one-element model, an item can be in one of two states: a learned state L and un unlearned state $\overline{L}$. If an item is in state L it is always responded to correctly. If it is in state $\overline{L}$ it is responded to correctly with probability g. The rule giving the state transformation is nondeterministic. If an item is in state $\overline{L}$ at the beginning of a stage and is presented, then it changes its state to L with probability c (where c remains constant throughout the procedure). Unlike the linear model, the one-element model is response sensitive. If an error is made in response to an item then that item was in state $\overline{L}$ at the time that the response was made. To see how this fact influences the response probability it is convenient to introduce a random variable X_n in the following way:

$$X_n^{(j)} = \begin{cases} 1, \text{ if an error occurs on the } n^{th} \text{ presentation of item } j \\ 0, \text{ if a success occurs on the } n^{th} \text{ presentation of item } j. \end{cases}$$

Then

$$\Pr(X_n^{(j)} = 1) = (1-g)(1-c)^{n-1} , \tag{4}$$

but

$$\Pr(X_n^{(j)} = 1 | X_{n-1}^{(j)} = 1) = (1-g)(1-c) . \tag{5}$$

In contrast, for the single-operator linear model

$$\Pr(X_n^{(j)} = 1) = \Pr(X_n^{(j)} = 1 | X_{n-1}^{(j)} = 1) = q_1^{(j)}\alpha^{n-1} . \tag{6}$$

Although these two models cannot be expected to provide the same optimization scheme in general, they are equivalent when only response insensitive strategies are considered. This is due to the fact that, if α is set equal to $1-c$ and $q_1^{(j)}$ to $1-g$ then identical expressions for $\Pr(X_n^{(j)} = 1)$ result for both models.

With the introduction of models such as these, the state space W and the transformation T can be defined in terms of some "learning state" of the student. For example, in the case of the linear model and a list of m stimulus items, we can define the state of the student at stage i as the m-tuple

$$q = (q^{(1)}, q^{(2)}, \cdots, q^{(m)}) \tag{7}$$

where $q^{(j)}$ denotes the current probability of making an error to item $s^{(j)}$ and define $T(q,s_j)$ as the vector obtained by replacing $q^{(j)}$ with $\alpha q^{(j)}$. This notation is illustrated in Fig. 3. If the behavioral optimization criterion were a test of the m items administered immediately

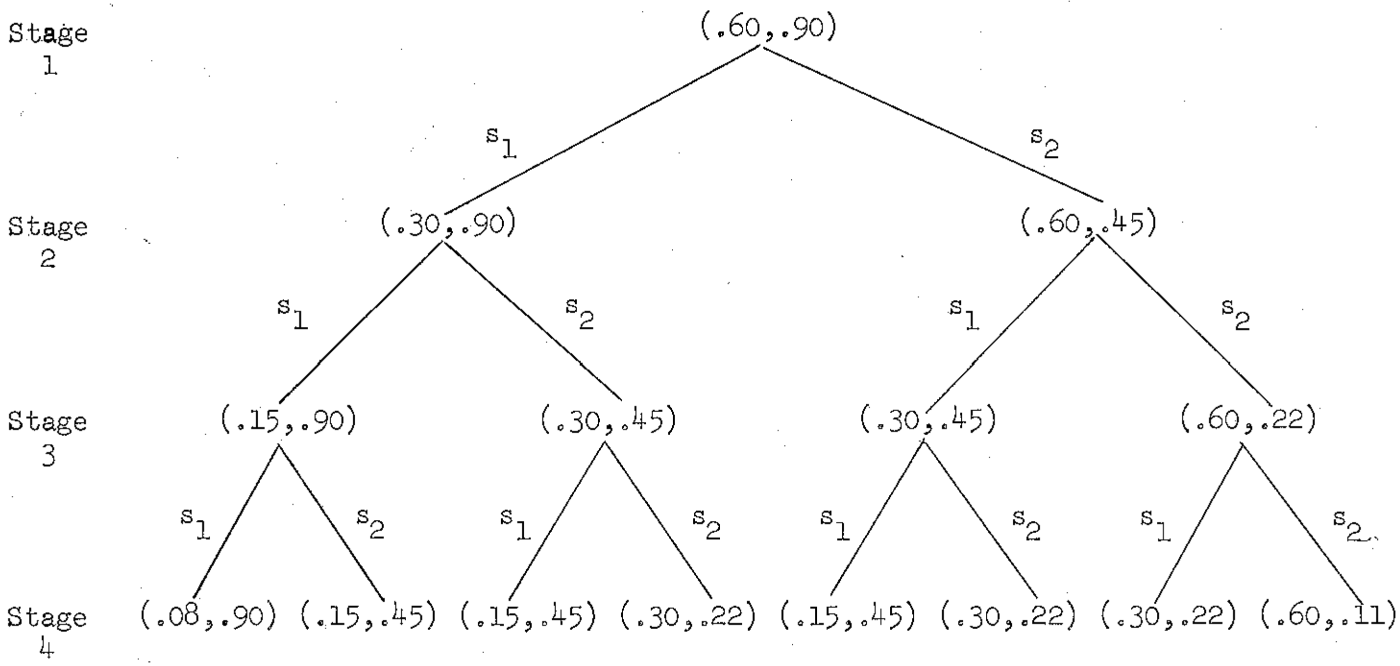

Figure 3. Outcome tree of response probabilities for linear model with $\alpha = 0.5$, $q_1^{(1)} = 0.6$, $q_1^{(2)} = 0.9$.

after stage N of the process, then the return function would be the expectation of the test score, i.e.,

$$\sum_{j=1}^{m} \left\{ 1 - q_N^{(j)} \right\}$$

where $1 - q_N^{(j)}$ is the probability of a correct response at the end of the instructional process. It is not necessary, however, that W be the actual state space of the model. It may, instead, be more convenient to define w_i as some function of the parameters of the model. For example, if the process of learning a list of simple items can be described by a one-element model, then w_i can be defined as the n-tuple whose j^{th} element is either L or $\overline{L}$. However, if one is interested in some criterion that can be expressed in terms of probabilities of the items being in state L, then it may be computationally more convenient to consider w_i as an n-tuple whose j^{th} element is the probability that stimulus $s^{(j)}$ is in state L at the beginning of stage i.

If we view the states as some function of the parameters of a learning model then the history h_i can be viewed as a suitable estimate of w_i. It is clear from our examples that a learning model can impose a severe constraint upon the history space in the sense that information regarding observable outcomes is rendered redundant. For example, if $q_1^{(j)}$ is known on a priori grounds (for each j), then the linear model renders the entire response history redundant. This is because the response probability of each item is completely determined by the number of times it has been presented. With the one-element model, the nature of the constraint on the history is not immediately clear. In general, the problem of deciding on an appropriate history, h_i, is similar to the problem of finding an observable statistic that provides a good estimate of a parameter. The history h_i may be regarded as an estimate of the state w_i. A desirable property for such a statistic would be for it to summarize all information concerning the state so that no other history would provide additional information. A history with this property can be called a sufficient history.

In the theory of statistical inference, a statistic with an analogous property is called a sufficient statistic. Since w_i is a function of the parameters of the model it would seem reasonable to expect that, if a sufficient statistic exists, for these parameters, then a sufficient history would be some function of the sufficient statistic. For a general discussion of the role of sufficient statistics in reducing the number of paths that must be considered in trees resulting from processes similar to those considered here, the reader is referred to Raiffa and Schlaiffer (1961).

Optimization Techniques

Up to now, the only technique we have considered that enables us to find an optimal strategy is to enumerate every path of the tree generated by the N-stage process. Although the systematic use of learning models can serve to reduce the number of paths that must be considered, much too large a number of paths still remains in most problems where a large number of different stimuli are used. The main success with a direct approach has been in the case of response insensitive strategies (Suppes, 1964; Crothers, 1965,1966; Dear, 1964). In these cases, either the number of stimulus types is drastically limited or the problem is of a type where the history can be simplified on a priori grounds. The techniques used in these approaches are too closely connected with the specific problem treated and the models used for any general discussion of their merits.

The theory of dynamic programming provides a set of techniques that reduce the portion of a tree that must be searched. These techniques have the merit of being model free. Moreover, they provide a computational algorithm which may be used to discover optimal strategies by numerical methods in cases where analytic methods are too complicated. The first application of dynamic programming to the design of optimal instructional systems was due to Smallwood (1962). Since then several other investigators have applied dynamic programming techniques to instructional problems of various types (Matheson, 1964; Dear, 1964; Karush and Dear, 1966). The results obtained by these investigators are too specific to be reviewed in detail. The main aim in this section is to indicate the nature of the techniques and how they can be applied to instructional problems.

Broadly speaking, dynamic programming is a method for finding an optimal strategy by systematically varying the number of stages and obtaining an expression which gives the return for a process with N stages as a function of the return from a process with $N-1$ stages. In order to see how this is done it is necessary to impose a restriction on the return function and define a property of optimal policies. Following Bellman (1961, p. 54) a return function is Markovian if, for any $K < N$, the effect of the remaining $N-K$ stages of the N-stage process upon the return depends only upon: 1) the state of the system at the end of the K^{th}

decision, and 2) whatever subsequent decisions are made. It is clear that the return function $\varphi(w_N)$ possesses this property. Another type of return function that possesses this property is one of the form:

$$g(w_1,d_1) + g(w_2,d_2) + \ldots + g(w_{N-1},d_{N-1}) + \varphi(w_N) .$$

A return function of this latter form may be important when cost as well as final test performance is an important criterion in designing the system. For example, in a computer-based system, $g(w_i,d_i)$ might be the cost of using the computer for the amount of time required to make decision d_i, and present the appropriate stimulus. Since the expressions resulting from a function of this form are somewhat more complicated, we will limit our attention to return functions of the form $\varphi(w_N)$. However, it should be borne in mind that essentially the same basic procedures can be used with the more complicated return function.

If a deterministic decision process has this Markovian property then an optimal strategy will have the property expressed by Bellman in his optimality principle : _whatever the initial state and the initial decision are, the remaining decisions constitute an optimal policy with regard to the state resulting from the first decision_ (Bellman, 1961, p. 57). To see how this principle can be utilized, let $f_N(w)$ denote the return from an N-stage process with initial state w if an optimal strategy is used throughout, and let us assume that T is deterministic and W is discrete. Since the process is deterministic, the final state is completely determined by the initial state w and the sequence of decision $d_1, d_2, \ldots, d_{N-1}$ (it should be recalled that no decision takes place during the last stage). If D_N denotes an arbitrary sequence of $N-1$ successive decisions (D_1

being the empty set) then the final state resulting from w and D_N can be written as $w'(D_N,w)$. The problem that must be solved is to find the sequence D_N which maximizes $\varphi[w'(D_N,w)]$. If such a sequence exists, then

$$f_N(w) = \max_{D_N} \varphi[w'(D_N,w)] \ . \tag{8}$$

While the solution of such a problem can be extremely complicated for arbitrary N, it is easily shown that for $N = 1$

$$f_1(w) = \varphi(w) \ . \tag{9}$$

As a result, if a relation can be found that connects $f_i(w)$ with $f_{i-1}(w)$ for each $i \leq N$ then $f_N(w)$ can be evaluated recursively by evaluating $f_i(w)$ for each i. Suppose that, in an i-stage process, an initial decision d is made. Then w is transformed into a new state, $T(w,d)$, and the decisions that remain can be viewed as forming an (i-1)-stage process with initial state $T(w,d)$. The optimality principle implies that the maximum return from the last $i - 1$ stages will be $f_{i-1}[T(w,d)]$. Moreover if $D_i = (d,d_2,\cdots,d_{i-1})$ and $D_{i-1} = (d_2,d_3,\cdots, d_{i-1})$ then

$$\varphi[w'(D_i,w)] = \varphi[w'(D_{i-1},T(w,d))] \ . \tag{10}$$

Suppose that D_{i-1} is the optimal strategy for the $i - 1$ stage process. Then the right-hand side of this equation is equal to $f_{i-1}[T(w,d)]$. An optimal choice of d is one which maximizes this function. As a result, the following basic recurrence relation holds:

$$f_n(w) = \max_d f_{n-1}[T(w,d)] \qquad 2 \leq n \leq N \tag{11}$$

$$f_1(w) = \varphi(w) \ . \tag{12}$$

Equations 11 and 12 relate the optimal return from an n-stage process with the optimal return from a process with only $n-1$ stages. Formally, n may be viewed as indexing a sequence of processes. All processes are identical except in the number of stages they possess. Thus, the solution of these equations provides us with a maximum return function $f_n(w)$ for each process and (also for each process) an initial decision d which ensures that this maximum will be attained if optimal decisions are made thereafter. It is important to note that both d and $f_n(w)$ are functions of w and that w should, in general, range over all values in the state space W. In particular, the initial state and initial decision of a typical member of the sequence of processes we are considering should not be confused with the initial state and initial decisions of the N-stage process we are trying to optimize. In fact, the initial decision of the 2-stage process corresponds to the last decision d_{N-1} of the N-stage process; the initial decision of the 3-stage process corresponds to the next to the last decision d_{N-2} of the N-stage process and so on.

The linear model of Eq. 2, with the state space defined in Eq. 7 provides an example of a deterministic process. The use of Eqs. 11 and 12 to find an optimal strategy for the special case of a 4-stage process with two items is illustrated in Table 1.

The state at the beginning of stage i is defined by the vector $(q_i^{(1)}, q_i^{(2)})$. The optimization criterion is the score on a test administered at the end of the instructional process. Since item j will be responded to correctly with probability $1 - q_N^{(j)}$, the terminal return function for an N-stage process is $2 - (q_N^{(1)} + q_N^{(2)})$. The calculation is begun by

Table 1

Calculation of Optimal Strategy for Example of Figure 3 Using Dynamic Programming

Number of Stages in Process N	Initial State w	Initial Decision d	Next State T(w,d)	Final Return of Optimal N-1 Stage Process $f_{N-1}[T(w,d)]$	Optimal Decision
1	(.08,.90)			1.02	
	(.15,.45)			1.40	
	(.30,.22)			1.48	
	(.60,.11)			1.29	
2	(.15,.90)	1	(.08,.90)	1.02	2
		2	(.15,.45)	1.40	
	(.30,.45)	1	(.15,.45)	1.40	2
		2	(.30,.22)	1.48	
	(.60,.22)	1	(.30,.22)	1.48	1
		2	(.60,.11)	1.29	
3	(.30,.90)	1	(.15,.90)	1.40	2
		2	(.30,.45)	1.48	
	(.60,.45)	1	(.30,.45)	1.48	1 or 2
		2	(.60,.22)	1.48	
4	(.60,.90)	1	(.30,.90)	1.48	1 or 2
		2	(.60,.45)	1.48	

Optimal Strategies

	Stage 1	Stage 2	Stage 3
1.	Item 1	Item 2	Item 2
2.	Item 2	Item 1	Item 2
3.	Item 2	Item 2	Item 1

viewing the fourth stage as a 1-stage process and obtaining the return for each possible state by means of Eq. 12. The possible states at this fourth stage are obtained from Fig. 3. The third and fourth stages are then viewed as a 2-stage process and Eq. 11 is used to determine the return that results from presenting each item for every possible state that can occur in stage 3, the previously computed result for a 1-stage process being used to complete the computations. For each state, the item with the maximum return represents the optimal decision to make at stage 3. The 3-stage process beginning at stage 2 is analyzed in the same way, using the previously computed results for the last two stages. The result is an optimal decision at stage 2 for each possible state, assuming optimal decisions thereafter. Finally, the procedure is repeated for the 4-stage process beginning at stage 1. The optimal strategies of item presentation that result from this procedure are given at the bottom of Table 1.

With a nondeterministic process, the situation is considerably more complicated. The transformation T is some type of probability distribution and the final return is a mathematical expectation. While arguments based on the optimality principle allow one to obtain recursive equations similar in form to (11) and (12), both the arguments used to obtain the equations and the methods used to solve them can contain many subtle features. A general review of the problems encountered in this type of process is given by Bellman (1962) and some methods of solution are discussed by Bellman and Dreyfus (1962). For the case where the transformation defines a Markov process with observable states, Howard (1960) has derived a set of equations together with an iterative technique of solution which has quite general applicability. However, in the case of instructional processes, it has so far tended to be the case that either the learning model used has unobservable states or that the process can be reduced to a more deterministic one (as is the case with the linear model discussed in the example above).

A response-insensitive process can often be viewed as a deterministic process. This is not, in general, possible with a response-sensitive process. The only process of this type that has been extensively analyzed is that in which a list of stimulus-response items is to be learned, the return function is the score on the test administered at the end of the process, and the learning of each item is assumed to occur independently and obey the assumptions of the one-element model. An attempt to solve this problem by means of a direct extension of Howard's techniques to Markov processes with unobservable states has been made by Matheson (1964). However, this approach appears to lead to somewhat cumbersome equations that are impossible to solve in any non-trivial case. A more promising approach has been devised by Karush and Dear (1966). As in our example of the linear model, the states of the process are defined in terms of the current probability that an item is in the conditioned state and a similar (though somewhat more general) return function is assumed. An expression relating the return from an (N-1)-stage process to the return from an N-stage process is then derived. The main complication in deriving this expression results from the fact that the outcome tree is more complicated, the subject's responses having to be explicitly considered. Karush and Dear proceed to derive certain properties of the return function and prove that in an

N-trial experiment[2] with items $s^{(1)}, s^{(2)}, \ldots, s^{(m)}$ (where $N \geq m$) and arbitrary initial conditioning probabilities $(\lambda^{(1)}, \lambda^{(2)}, \ldots, \lambda^{(m)})$, an optimal strategy is given by presenting at any trial an item for which the current conditioning probability is least. In most applications the initial probabilities $\lambda^{(j)}$ can be assumed to be zero. In this case, an observable sufficient history can be defined in terms of a counting process. An optimal strategy is initiated by presenting the m items in any order on the first m trials and a continuation of this strategy is optimal if and only if it conforms to the following rule:

1. For every item set the count at 0 at the beginning of trial $m+1$.
2. Present an item at a given trial if and only if its count is least among the counts for all items at the beginning of the trial.
3. Following a trial, increase the count for the presented item by 1 if the response was correct but set it at 0 if the response was incorrect.

General Discussion

In this paper we have attempted to achieve two main goals. The first has been to provide an explicit statement of the problems of optimal instruction in the framework of multi-stage decision theory. Our main reason for introducing a somewhat elaborate notational system is the need for a clear distinction between the optimization problem, the learning process that the student is assumed to follow, and the method of solving the optimization problem. The second goal has been to indicate, using dynamic programming as an example, how optimization problems can be solved in practice. Again, it should be emphasized that dynamic programming is not the only technique that can be used to solve optimization problems. Many response-insensitive problems are solvable by more simple, though highly specific, techniques. However, dynamic programming is the only technique that has so far proved useful in the derivation of response-sensitive strategies. In describing dynamic programming an attempt has been made to emphasize two basic features: the optimality principle, and the backward-induction procedure by means of which an optimal strategy is obtained by starting, in effect, at the last stage. It should be noted that these can be used independently. For example, it is possible to combine the optimality principle with a forward induction procedure which starts at the first atage of the process.

In any attempt to apply an optimization theory in practice one must ask the question; how can it be tested experimentally? In principle, it is easy to formulate such an experiment. A number of strategies are compared--some theoretically optimal, others theoretically suboptimal. A test is administered at the end of the process that is designed to be some observable function of the final return. However, the only experiment that has been explicitly designed to test an optimization theory is that by Dear, Silberman, Estavan, and Atkinson (1965), although in the case of response-insensitive theories, it is often possible to find experiments in the psychological literature which provide indirect support.

The experiment reported by Dear et al. was concerned with testing the strategy proposed by Karush and Dear (1966) for the case outlined in the preceding section. The major modification of this strategy was to prohibit repeated presentations of the same item by forcing separations of several trials between presentations of individual items.[3] Each subject was presented with two sets of paired-associate items. The first set of items was presented according to the optimization algorithm. Items in the second set were presented an equal number of times in a suitable random order. (It will be recalled that this strategy is optimal if the linear model is assumed.) It was found that while the acquisition data (e.g., rate of learning) tended to favor items in the first set, no significant difference was found in post-test scores between items of the two sets.

It follows from the result of this experiment that, even for a simple problem such as this, an optimization theory is needed that assumes a more complicated learning model. At least one reason for this is that, in simple paired-associate experiments that result in data which is fitted by the one-element model, any systematic effects of stimulus-presentation sequences are usually eliminated by presenting different subjects with different random sequences of stimuli. When a specific strategy is used, it may be the case that either the assumption of a forgetting process or of some short-term memory state becomes important in accounting for the data (Atkinson and Shiffrin, 1965).

Unfortunately, the analytic study of the optimization properties of more complex models, at least by dynamic programming techniques, is difficult. The only major extension of response-sensitive models has been a result of Karush and Dear (1965) which shows that the optimal strategy for the one-element model is also optimal if it is assumed that the probability of a correct response in the conditioned state L is less than one. However, there are ways by means of which good approximations to optimal strategies might be achieved, even in the case of extremely complex models. Moreover, in many practical applications, one is not really critically concerned about solving for an optimal procedure, but would instead be willing to use an easily determined procedure that closely approximates the return of the optimum procedure. The main means of achieving a good approximation is by analyzing the problem numerically, computing the optimal strategy for a large number of special cases. A useful general algorithm for doing this is the backward induction procedure described in the preceding section. Table 1 illustrates how this algorithm can be used to find an optimal strategy for one particular case. Dear (1964) discusses the use of this algorithm in other response-insensitive problems.

The chief disadvantage of the backward induction algorithm is that it can only be used for optimal strategy problems involving a fairly small number of stages. Although its use can eliminate the need to search every branch of a tree, the computation time still increases as a function of the number of possible final states that can result from a given initial state. However, a backward induction solution for even a small number of stages would provide a locally optimal policy for a process with a large number of stages, and this locally optimal strategy might provide a good approximation to an optimal strategy. To decide how "good" an approximation such a strategy provided, its return could be evaluated and this could be compared with the returns of alternative strategies.

References

Atkinson, R. C., Bower, G. H., and Crothers, E. J. *An introduction to mathematical learning theory*. New York: Wiley, 1965.

Atkinson, R. C., and Estes, W. K. Stimulus sampling theory. In R. D. Luce, R. R. Bush, and E. Galanter (Eds.) *Handbook of mathematical psychology*. Vol. 2. New York: Wiley, 1963. Pp. 121-268.

Atkinson, R. C., and Shiffrin, R. M. Mathematical models for memory and learning. Tech. Rep. 79, Institute for Mathematical Studies in the Social Sciences, Stanford University, 1965. (To be published in *The anatomy of memory*, *Vol. 3*, *Proceedings of the Third Conference on Learning, Remembering, and Forgetting*, edited by D. P. Kimble, Palo Alto, Calif.: Science and Behavior Books and Co., 1966.)

Bellman, R. *Dynamic programming*. Princeton: Princeton Univer. Press, 1957.

Bellman, R. *Adaptive control processes*. Princeton: Princeton Univer. Press, 1961.

Bellman, R. Functional equations. In R. D. Luce, R. R. Bush, and E. Galanter (Eds.) *Handbook of mathematical psychology*. Vol. 3. New York: Wiley, 1965. Pp. 487-513.

Bellman, R., and Dreyfus, S. E. *Applied dynamic programming*. Princeton: Princeton Univer. Press, 1962.

Bower, G. H. Application of a model to paired-associate learning. *Psychometrika*, 1961, 26, 255-280.

Bruner, J. S. Some theorems on instruction stated with reference to mathematics. In E. R. Hilgard (Ed.) *Theories of learning and theories of instruction*. 63rd NSSE Yearbook, Part 1, 1964. Pp. 306-335.

Bush, R. R., and Sternberg, S. H. A single operator model. In R. R. Bush and W. K. Estes (Eds.) Studies in mathematical learning theory. Stanford: Stanford Univer. Press, 1959. Pp. 204-214.

Cronbach, L. J., and Gleser, Goldine. Psychological tests and personnel decisions. Urbana: Univer. of Illinois Press, 1965.

Crothers, E. J. Learning model solution to a problem in constrained optimization. Journal of Mathematical Psychology, 1965, 2, 19-25.

Crothers, E. J. Optimal presentation sequences for items from several categories. Journal of Mathematical Psychology, 1966, in press.

Dear, R. E. Solutions for optimal designs of stochastic learning experiments. Tech. Rep. SP-1765/000/00, System Development Corp., Santa Monica, 1964.

Dear, R. E., and Atkinson, R. C. Optimal allocation of items in a simple, two-concept automated teaching model. In J. E. Coulson (Ed.) Programmed learning and computer-based instruction. New York: Wiley, 1962. Pp. 25-45.

Dear, R. E., Silberman, H. F., Estavan, D. P., and Atkinson, R. C. An optimal strategy for the presentation of paired-associate items. Tech. Memo. TM-1935/101/00, System Development Corp., Santa Monica, 1965.

Estes, W. K. Learning theory and the new mental chemistry. Psychological Review, 1960, 67, 207-223.

Gage, N.L. (Ed.) Handbook of research on teaching. Chicago: Rand McNally, 1963.

Hilgard, E. R. (Ed.) Theories of learning and theories of instruction. 63rd NSSE Yearbook, Part 1, 1964.

Howard, R. A. *Dynamic programming and Markov processes*. New York: Wiley, 1960.

Karush, W., and Dear, R. E. Optimal procedure for an N-state testing and learning process--II. Tech. Rep. SP-1922/001-00, System Development Corp., Santa Monica, 1965.

Karush, W., and Dear, R. E. Optimal stimulus presentation strategy for a stimulus sampling model of learning. *Journal of Mathematical Psychology*, 1966, 3, 19-47.

Lumsdaine, A. A. Instruments and media of instruction. In N. L. Gage (Ed.) *Handbook of research on teaching*. Chicago: Rand McNally, 1963. Pp. 583-683.

Matheson, J. Optimum teaching procedures derived from mathematical learning models. Report CC52, Institute in Engineering-Economic Systems, Stanford University, 1964.

Raiffa, H., and Schlaiffer, R. Applied statistical decision theory. Graduate School of Business Administration, Harvard Univer., 1961.

Silberman, H. F. Characteristics of some recent studies of instructional methods. In J. E. Coulson (Ed.) *Programmed learning and computer-based instruction*. New York: Wiley, 1962. Pp. 13-24.

Smallwood, R. D. *A decision structure for teaching machines*. Cambridge, Mass.: MIT Press, 1962.

Sternberg, S. H. Stochastic learning theory. In R. D. Luce, R. R. Bush, and E. Galanter (Eds.) *Handbook of mathematical psychology*. Vol. 2. New York: Wiley, 1963. Pp. 1-120.

Suppes, P. Problems of optimization in learning a list of simple items. In M. W. Shelly, II, and G. L. Bryan (Eds.). *Human judgment and optimality*. New York: Wiley, 1964, 116-126.

Footnotes

[1] Support for this research was provided by the National Aeronautics and Space Administration, Grant No. NGR-05-020-036, and by the Office of Education, Grant No. OE5-10-050.

[2] Here the term N-trial experiment refers to an anticipatory paired-associate procedure which involves N stimulus presentations. To each stimulus presentation the subject makes a response and then is told the correct answer for that stimulus.

[3] This modification was necessary because it has been shown experimentally that if the same item is presented on immediately successive trials then the subject's response is affected by considerations of short-term memory.