Department of Computer Science Technical Reports

Department of Computer Science

1974

The Algorithm Selection Problem—Abstract Models

John R. Rice

Purdue University, jrr@cs.purdue.edu

Report Number:

74-116

THE ALGORITHM SELECTION PROBLEM - ABSTRACT MODELS

John R. Rice
Computer Science Department
Purdue University
West Lafayette, Indiana 47907

May 1974

CSD-TR 116

1. INTRODUCTION. The problem of selecting an effective or good or best algorithm arises in a wide variety of situations. The context of these situations often obscures the common features of this selection problem and the purpose of this report is to formulate abstract models appropriate for considering it. Within the framework established by these models we present a variety of questions that can (and usually should) be asked in any specific application.

It should be made clear that we do not believe that these models will lead directly (by simple specialization) to superior selection procedures. This will always require exploitation of the specific nature of the situation at hand. Even so, we do believe that these models will clarify the consideration of this problem and, in particular, show that some approaches used are based on naive assumptions about the selection assumption.

This is the first of a series of reports which consider the following topics:

Abstract Models

Concrete Examples

Numerical Analysis - Selection of Quadrature Algorithms

Operating Systems - Selection of Scheduling Algorithms

Artificial Intelligence - Learning Algorithms

Approximation Theory for Selection Procedures

Computation of Selection Procedures

The three concrete examples which the reader can use to interpret the abstractions in this report may be summarized as follows:

Quadrature: One is given a function $f(x)$, an interval $[a,b]$ and a tolerance $\in > 0$. One is to select an algorithm to estimate

$$\int_a^b f(x)dx$$

which is efficient (uses few evaluations of $f(x)$) and reliable (produces an estimate within the specified tolerance).

Operating Systems: One is given an environment for a large computer operation. Information known includes the mix of jobs between batch, interactive and semi-interactive, some basic characteristics of these classes of jobs and the characteristics of the computer operation. One is to select an algorithm to schedule the execution of these jobs which produce (a) high batch throughput, (b) good response to interactive jobs, (c) good service to semi-interactive jobs and (d) high priority fidelity.

Artificial Intelligence: One is given a description of the game Tic-Tac-Toe. One is to select an algorithm to play the game which is effective, i.e. never loses and wins whenever an opponent's mistake allows it.

A selection procedure is invariably obtained by assigning values to parameters in general "form". More precisely, the selection procedure itself is an algorithm and a specific class is chosen with free parameters and these parameters are then chosen so as to satisfy (as well as they can) the objectives of the selection problem. Classical forms include things like polynomials (with coefficients as parameters) and linear mappings (with matrix coefficients or weights as parameters). Other relevant forms are decision trees (with size, shape and individual decision elements as parameters) and programs (with various program elements as parameters).

The models presented here are primarily aimed at algorithm selection problems with the following three characteristics:

Problem Space: The set of problems involved is very large and quite diverse. This set is of high dimension in the sense that there are a number of independent characteristics of the problems which are important for the algorithm selection and performance. There is usually considerably uncertainty about these characteristics and their influences.

Algorithm Space: The set of algorithms that needs to be considered is large and diverse. Ideally there may be millions of algorithms and practically there may be dozens of them. In counting algorithms we do not distinguish between two which are identical except for the value of some numeric parameter. Again this set is of high dimensions and there is uncertainty about the influence of algorithm characteristics.

Performance Measure: The criteria to measure the performance of a particular algorithm for a particular problem are complex and hard to compare (e.g. one wants fast execution, high accuracy and simplicity). Again there is considerable uncertainty in assigning and interpreting these measures.

2. THE BASIC MODEL. We describe the basic abstract model by the diagram in Figure 1. The items in this model are defined below in detail so as to be completely clear about the nature of the model.

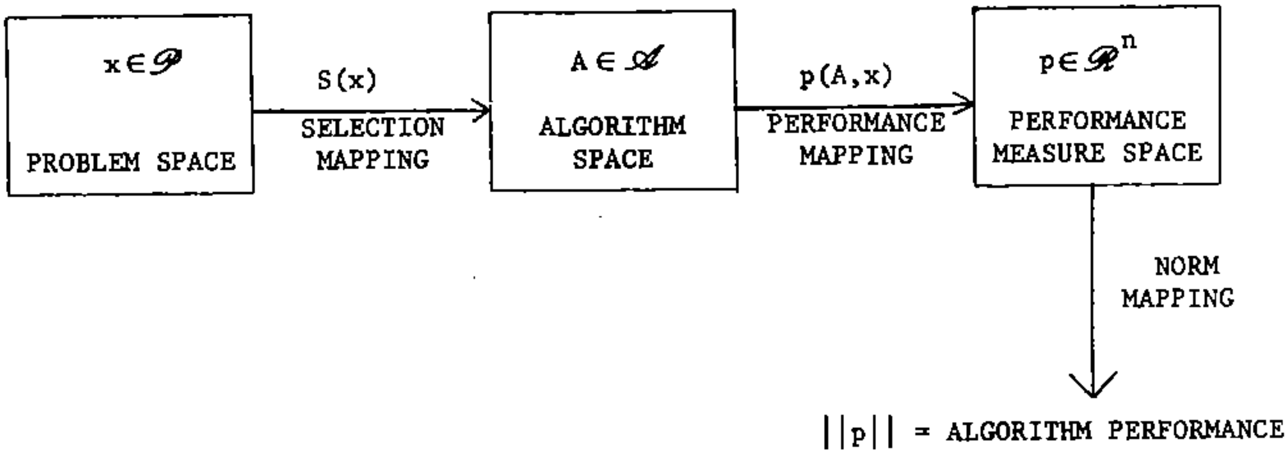

Figure 1. Schematic diagram of the basic model for the algorithm selection problem. The objective is to determine S(x) so as to have high algorithm performance.

Definitions for the basic Model:

$\mathscr{P}$ = Problem space or collection

x = Member of $\mathscr{P}$, problem to be solved

$\mathscr{A}$ = Algorithm space or collection

A = Member of $\mathscr{A}$, algorithm applicable to problems from $\mathscr{P}$

S = Mapping from $\mathscr{P}$ to $\mathscr{A}$

$\mathscr{R}^n$ = n-dimensional real vector space of performance measures

p = Mapping from $\mathscr{A} \times \mathscr{P}$ to $\mathscr{R}^n$ determining performance measures

$|| \;\; ||$ = Norm on $\mathscr{R}^n$ providing one number to evaluate an algorithm's performance on a particular problem.

For completeness we now state the:

Algorithm Selection Problem: Given all the other items in the above model, determine the selection mapping $S(x)$.

There must be, of course, some criteria for this selection and we present four primary ones below:

A. Best Selection. Choose that selection mapping $B(x)$ which gives maximum performance for each algorithm:

$$||p(B(x),x)|| \geq ||p(A,x)|| \qquad \text{for all } A \in \mathcal{A}$$

B. Best Selection for a Subclass of Problems. One is to choose just one algorithm to apply to every member of a subclass $\mathcal{P}_0 \subset \mathcal{P}$. Choose that selection mapping $S(x) = A_0$ which minimizes the performance degradation for members of $\mathcal{P}_0$ (compared to choosing $B(x)$):

$$\max_{x \in \mathcal{P}_0} [||p(B(x),x)|| - ||p(A_0),x)||] \leq \max_{x \in \mathcal{P}_0} [||p(B(x),x)|| - ||p(A,x)||]$$

$$\text{for all } A \in \mathcal{A}$$

C. Best Selection from a Subclass of Mappings. One is to restrict the mapping $S(x)$ to be of a certain form or from a certain subclass $\mathcal{S}_0$ of all mapping from $\mathcal{P}$ to $\mathcal{A}$. Choose that selection mapping $S^*(x)$ from $\mathcal{S}_0$ which minimizes the performance degradation for all members of $\mathcal{P}$.

$$\max_{x \in \mathcal{P}} [||p(B(x),x)|| - ||p(S^*(x),x)||] \leq \max_{x \in \mathcal{P}} [||p(B(x),x)|| - ||p(S(x),x)||]$$

$$\text{for all } S \in \mathcal{S}_0$$

D. Best Selection from a Subclass of Mappings and Problems. One is to choose just one algorithm from a subclass $\mathcal{S}_0$ to apply every member of a subclass $\mathcal{P}_0 \subset \mathcal{P}$. Choose that selection mapping $S^*(x)$ from $\mathcal{S}_0$ which minimizes the performance degradation for all members of $\mathcal{P}_0$:

$$\max_{x \in \mathcal{P}_0} [||p(B(x),x || - ||p(S^*(x),x)||] \leq \min_{S \in \mathcal{S}_0} \max_{x \in \mathcal{P}_0} [||p(B(x),x)|| - ||p(S(x),x)||]$$

These four criteria do not exhaust the meaningful criteria but they do illustrate the principal ideas. There are five main steps to the analysis and solution of the algorithm selection problem as follows

<u>Step 1</u> (Formulation) Determination of the subclasses of problems and mappings to be used.

<u>Step 2</u> (Existence) Does a best selection mapping exist?

<u>Step 3</u> (Uniqueness) Is there a unique best selection mapping?

<u>Step 4</u> (Characterization) What properties characterize the best selection mapping and serve to identify it?

<u>Step 5</u> (Computation) What methods can be used to actually obtain the best selection mapping.

The reader familiar with the theory of approximation of functions will observe that this framework is familiar and that we may put that classical theory within this framework. The space $\mathscr{P}$ is a function space and the algorithm space $\mathscr{A}$ may be identified with a subspace of $\mathscr{P}$. The algorithm enters as the means of evaluating elements of $\mathscr{A}$. The performance mapping is

$$p(A,x) = ||x(t) - A(t)||_{\mathscr{P}}$$

where the norm is taken on $\mathscr{P}$. Thus the performance measure space is $\mathscr{R}^1$ and the norm mapping is trivial.

There are two remarks needed about this observation. First, the body of significant material in approximation theory is large. It would require, no doubt, from 2000 to 4000 pages to present a reasonably complete and concise exposition of the results currently known. This implies that there is a very rich body of material waiting to be applied to the algorithm selection problem, either directly or by analogy. Second, and more important, the algorithm selection problem is an essential extension and generalization of approximation theory. We will see concrete examples of this problem where the current theory of approximation has nothing relevant to apply except by the faintest analogies.

We next present two concrete examples to illustrate this model:

Example 1: (A Quadrature problem). Given $f(x) \in C^4[0,1]$ and $\in > 0$, estimate $\int_0 x(t)dx$ within $\in$ by a composite Newton-Cotes formula of degree k and with ℓ points with a minimum number of evaluations of f(x). We see that

$\mathscr{P} = C^4[0,1] \times \mathscr{R}^1$

$\mathscr{A} = I^2$ (where I denotes the positive integers)

n = 1 in the performance measure space

We choose two subclasses:

$\mathscr{P}_0 = \{x(t) | x(t)$ has at most 1 inflection point$\}$

$\mathscr{S}_0$ = linear function of $\in$, x(0), x(1/3), x(2/3) and x(1).

Thus S(x) would have the general form with 10 parameters

$$S(x(t)) = \begin{pmatrix} s_{11} & s_{12} & s_{13} & s_{14} & s_{15} \\ s_{21} & s_{22} & s_{23} & s_{24} & s_{25} \end{pmatrix} \quad (x(0), x(1/3), x(2/3), x(1), \in)^T$$

or perhaps one might choose the more restricted form

$$S(x(t)) = \begin{pmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{pmatrix} \begin{pmatrix} \sum_{i=0}^{3} \alpha_i x(t/3) \\ \in \end{pmatrix}$$

which has only 8 parameters.

Example 2: (As game playing problem). We are to devise an algorithm for playing Tic-Tac-Toe. The problem space is the set of partial games of Tic-Tac-Toe. While this number is large, there are in fact only 28 distinct reasonable games if one eliminates blunders, symmetries and board rotations. The space $\mathscr{A}$ may be represented as a space of large tables of responses for each situation. However, we restrict our selection to a decision tree that involves only the existence of immediate winning positions and vacant position types. The algorithm form may then be represented as shown in Figure 2. There are 30 parameters S_{ij} in this form of the selection mapping which take on the values "yes" or "no". Only 15 of these are independent. In addition there are 16 parameters a_i which take on one of the following five values.

1. Play the winning move
2. Block the opponent's win
3. Play in the center square
4. Play in a corner (first free one clockwise from upper right)
5. Play in a side (first free one clockwise from right)

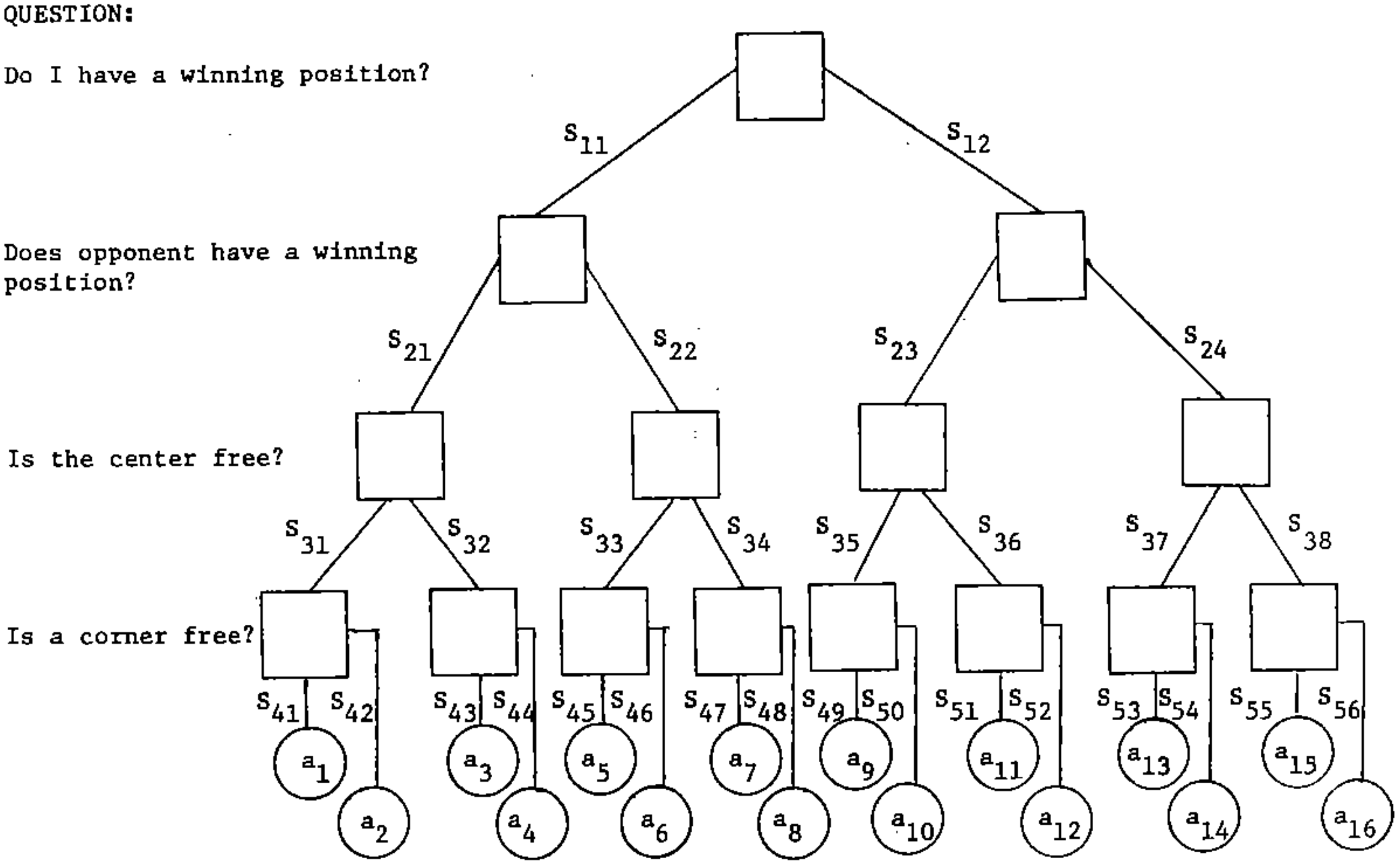

Figure 2. The form of the selection mapping for the Tic-Tac-Toe example. Each S_{ij} is a "yes" or "no" and each a_i is one of five moves.

An examination of this game shows that we have been overly elaborate here. Thus we may assign S_{11} = "yes" and S_{12} = "no" and then a_i = Move 1 for $i = 1,2,\ldots,8$ is certainly called for. However, it is still of interest to reflect upon how one would compute this if one had no a priori information about the game.

3. THE MODEL WITH SELECTION BASED ON FEATURES. An examination of various instances of the algorithm selection problem shows that there is another ingredient almost always present. It is sometimes explicit and sometimes not and we call this selection based on features of the problem. This model is described by the diagram in Figure 3.

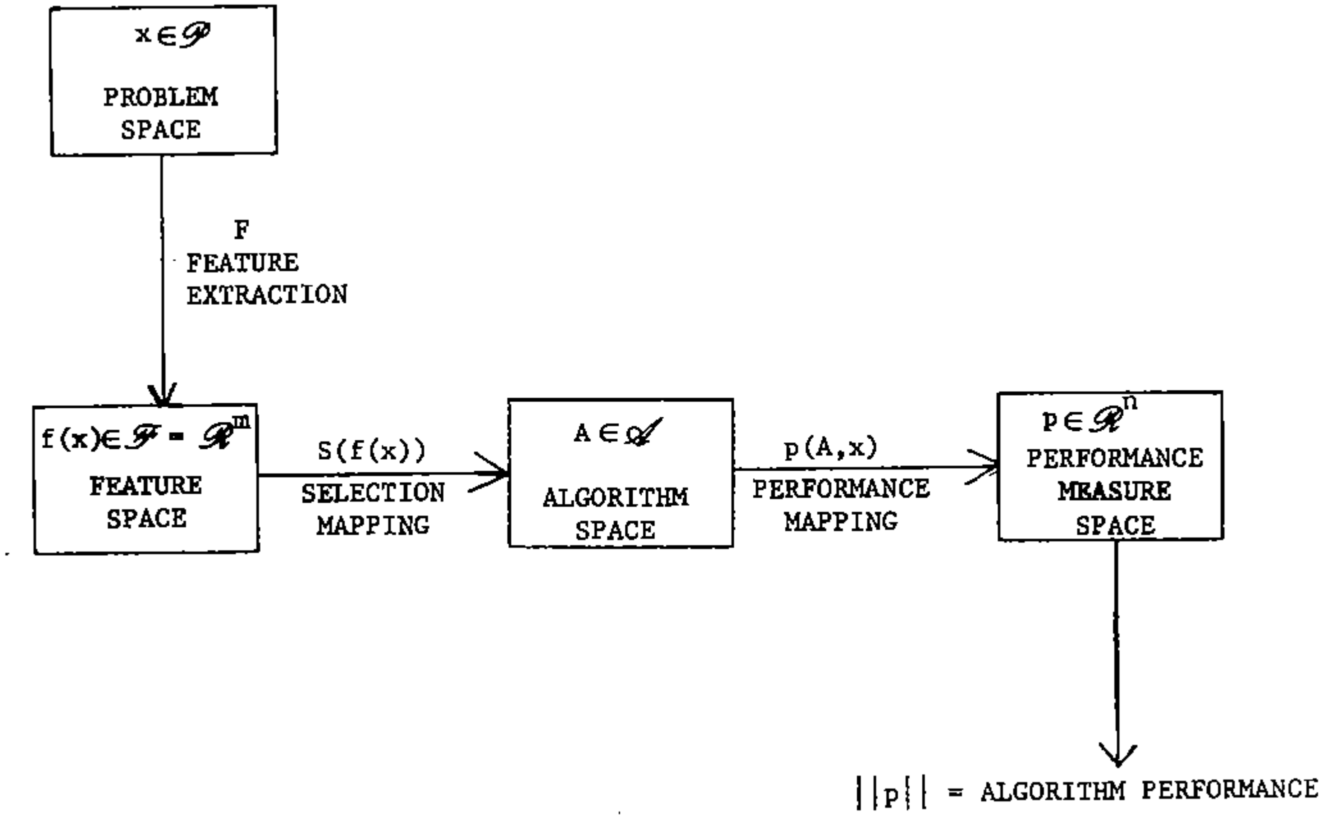

Figure 3. Schematic diagram of the model with selection based on features of the problem.

The additional definitions for this model are:

$\mathscr{F}$= Feature space identified with $\mathscr{R}^m$ here to suggest it is simpler and of lower dimension than $\mathscr{P}$.

F = Mapping from $\mathscr{P}$ to $\mathscr{F}$ which associates features with problems.

Note that the selection mapping now depends only on the features f(x) but yet the performance mapping still depends on the problem x. The introduction of features may be viewed as a way to systematize the introduction of problem subclasses in the basic model.

The previous statement of the algorithm selection problem and the criteria for selection are still valid for this new model as well as the five steps in the analysis and solution of the problem. The determination of the features to be used is frequently part of the selection process, often one of the most important parts. One may view the features as an attempt to introduce an approximate coordinate system in $\mathcal{P}$. Ideally, those problems with the same features would have the same performance for any algorithm being considered. Since this ideal is rarely achieved, we may pose several specific questions about the determination of features.

E. <u>Best Features for a Particular Algorithm</u>. Given an algorithm A and the dimension m of $\mathcal{F}$, what m features are the best for the prediction of the performance of A. Let $\mathcal{S}(f)$ denote the equivalence class of all those problems $x,y\in\mathcal{P}$ so that $F(x) = F(y) = f$. We then wish to determine the mapping F^* and associated equivalence classes $\mathcal{S}^*(f)$ so that

$$d_m^*(A) = \max_{f\in\mathcal{F}} \max_{x,y\in\mathcal{S}^*(f)} ||p(A,x) - p(A,y)|| \leq \max_{f\in\mathcal{F}} \max_{x,y\in\mathcal{F}(f)} ||p(A,x) - p(A,y)||$$

The selection of best features corresponds to the selection of best approximating subspaces in approximation theory and leads one to ideas of n-widths and entropy of the problem space $\mathcal{P}$. Roughly speaking, if d_m^* is large then the effective dimension of $\mathcal{P}$ (for the problem at hand) is probably much larger than m and, conversely, if d_m^* is small then the effective dimension of $\mathcal{P}$ is close to m.

F. <u>Best Features for a Class of Algorithms</u>. Given a set $\mathcal{A}_0 \subset \mathcal{A}$ and the dimension m of $\mathcal{F}$, what m features are the best for prediction of the performance of algorithm $A\in\mathcal{A}_0$? With the previous notation we wish to determine F^* and $\mathcal{S}^*(f)$ so that

$$d_m^*(\mathscr{A}_0) = \max_{f\in\mathscr{F}} \max_{A\in\mathscr{A}_0} \max_{x,y\in\mathscr{G}^*(f)} ||p(A,x) - p(A,y)||$$

$$\leq \max_{f\in\mathscr{F}} \max_{A\in\mathscr{A}_0} \max_{x,y\in\mathscr{G}(f)} ||p(A,x) - p(A,y)||$$

G. Best Features for a Subclass of Selection Mappings. Given a subclass $\mathscr{S}_0$ of selection mappings from $\mathscr{F}$ to $\mathscr{A}$, what m features are the best for prediction of the performance of algorithms? With the previous notation we wish to determine F^* and $\mathscr{G}^*(f)$ so that

$$d_m^*(\mathscr{S}_0) = \max_{f\in\mathscr{F}} \max_{S\in\mathscr{S}_0} \max_{x,y\in\mathscr{G}^*(f)} ||p(S(f),x) - p(S(f),y)||$$

$$\leq \max_{f\in\mathscr{F}} \max_{S\in\mathscr{S}_0} \max_{s,y\in\mathscr{G}(f)} ||p(S(f),x) - p(S(f),y)||$$

The determination of the best (or even good) features is one of the most important, yet nebulous, aspects of the algorithm selection problem. Many problem spaces $\mathscr{P}$ are known only in vague terms and hence an experimental approach is often used to evaluate the performance of algorithms over $\mathscr{P}$. That is, one chooses a sample from $\mathscr{P}$ and restricts considerations to this sample. An appropriate sample is obviously crucial to this approach and if one has a good set of features for $\mathscr{P}$, then one can at least force the sample to be representative with respect to these features. Note that the definition of best features is such that they are the items of information most relevant to the performance of algorithms for the problem at hand.

In some well understood areas of computation there is a generally agreed upon (if not explicitly stated) set of features. For example, consider the problem of solving a linear system Ax = b of equations. The features include descriptors like: small order, sparse, band, diagonally dominant, positive definite, ill-conditioned, etc. Given values for these features an experienced numerical analyst can select an appropriate algorithm for this problem with considerable confidence. The selection problem for quadrature is already much more difficult and the solution of simultaneous systems of nonlinear equations is very poorly understood. If this situation exists for problems that have been studied for one or two centuries then one should not be surprised by the difficulties and uncertainties for problems that have just appeared in the past one or two decades.

4. ALTERNATE DEFINITIONS OF BEST FOR THE MODELS. In the preceding sections we have uniformly taken a minimax approach to the definition of best or optimum selection. That is, we have minimized the effect of the worst case. It is reasonable to ignore the performance for the worst case and, instead, consider optimizing some sort of average behavior. In this section we exhibit the resulting mathematical problems corresponding to using a least squares or least deviation approach (these correspond to L_2 and L_1 optimization in mathematical terms). We have identified seven problems label A through G. Problem A is unaffected by these considerations so let us consider Problem B: Best Selection for a Subclass of Problems. We use the notation introduced with the original mathematical statement of this problem which is:

Minimax Approach

$$\max_{x\in\mathcal{P}_0} [||p(B(x),x)|| - ||p(A^*,x)||] \leq \max_{x\in\mathcal{P}_0} [||p(B(x),x)|| - ||p(A,x)||]$$

for all $A\in\mathcal{A}$

The corresponding mathematical statements for the least squares and least deviation approach are:

Least Squares Approach

$$\int_{\mathcal{P}_0} [||p(B(x),x)|| - ||p(A^*,x)||]^2 dx \leq \int_{\mathcal{P}_0} [||p(B(x),x)|| - ||p(A,x)||]^2 dx$$

for all $A\in\mathcal{A}$

Least Deviations Approach

$$\int_{\mathcal{P}_0} \Big| ||p(B(x),x)|| - ||p(A^*,x)|| \Big| dx \leq \int_{\mathcal{P}_0} \Big| ||p(B(x),x)|| - ||p(A,x)|| \Big| dx$$

for all $A\in\mathcal{A}$

The use of integrals in these formulations implies that a topology has been introduced in the problem space $\mathcal{P}$. Many common examples for $\mathcal{P}$ are discrete in nature and in these cases the topology introduced reduces the integrals to sums. This technicality is unlikely to cause real difficulties and we continue to use integrals as this gives the neatest formulations. Note that the only difference between the two new formulations is the exponent (2 or 1) in the integrand. Thus we may avoid repeating these formulations twice by making this a variable, say r, which has values 1 or 2. Note that in approximation theory it is shown that minimax is the limiting case as $r \to \infty$ so that all three approaches can be expressed in one formulation with r as a parameter.

Recall that Problem C is the Best Selection from a Subclass of Mappings. The alternative mathematical formulation of this problem is

$$\int_{\mathscr{P}} \Big| ||p(B(x),x)|| - ||p(S_0(x),x)|| \Big|^r dx \le \int_{\mathscr{P}} \Big| ||p(B(x),x)|| - ||p(S(x),x|| \Big|^r dx$$

$$\text{for all } S \in \mathscr{S}_0$$

The alternative formulation for Problem D is identical to this except that the problem subclass $\mathscr{P}_0$ replace $\mathscr{P}$ as the domain of integration.

The next three problems involve features and we choose to use a consistent approach for the reformulations. That is, if we use least squares on the problem space we also use it on the feature space $\mathscr{F}$ and the algorithm space $\mathscr{A}$. If we set

$$d_m^r(A,\mathscr{S}) = \int_{f \in \mathscr{F}} \Big[\iint_{x,y \in \mathscr{S}(f)} ||p(A,x) - p(A,y)||^r \Big]^{1/r}$$

then for Problem E: Best Feature for a Particular Algorithm, the objective is to find the feature mapping F^* and associated equivalence classes $\mathscr{S}^*(f)$ which minimize $d_m^r(A,\mathscr{S})$ i.e.

$$d_m^r(A) = d_m^r(A,\mathscr{S}^*) = \min_{\mathscr{S}} d_m^r(A,\mathscr{S})$$

For Problem F we introduce

$$d_m^r(\mathscr{A}_0,\mathscr{S}) = \int_{f \in \mathscr{F}} \Big[\int_{A \in \mathscr{A}_0} \iint_{x,y \in \mathscr{S}(f)} ||p(A,x) - p(A,y)||^r \Big]^{1/r}$$

and then the objective is to determine F^* and associated $\mathscr{S}^*(f)$ so that

$$d_m^r(\mathscr{A}_0) = d_m^r(\mathscr{A}_0, \mathscr{S}^*) = \min_{\mathscr{S}} d_m^r(\mathscr{A}_0, \mathscr{S})$$

A similar approach to Problem G yields a similar expression except that the integral over $\mathscr{A}_0$ is replaced by an integral over $\mathscr{S}_0$.

In many practical problems there is little to guide one in the choice of a particular formulation of the mathematical optimization problem, i.e. should we choose $r = 1, 2$ or ∞? These choices might not be particularly significant in the larger context, but they are very significant in determining the difficulty of the resulting mathematical optimization problem. A lesson learned from practical approximation theory might be applicable in this larger context. This lesson is, roughly, that the crucial ingredients for success are proper choices of the subclasses $\mathscr{P}_0$, $\mathscr{A}_0$ and $\mathscr{S}_0$. Once these are made properly then the mathematical optimization should be made for that value of r that gives the least difficulty. If the problem is completely linear then $r = 2$ (least squares) almost always results in the least mathematical difficulty. The situation is variable for nonlinear problems. Note that there are practical approximation problems where the choice of r is crucial and no doubt there are similar cases for the algorithm selection problem. We are saying that the choice of r is important only in an infrequent number of instances.

5. THE MODEL WITH VARIABLE PERFORMANCE CRITERIA. We have assumed so far that there is a fixed way to measure the performance of a particular algorithm for a particular problem. There are, however, many situations where it is reasonable to view the performance criteria as input to the selection problem. Consider, for example, the selection of a program to solve ordinary differential equations and the criteria of speed, accuracy, reliability, and care of use. In different situations the weight given to each of these might vary from almost zero to almost 100%. A model for this version of the selection problem is shown in the diagram of Figure 4.

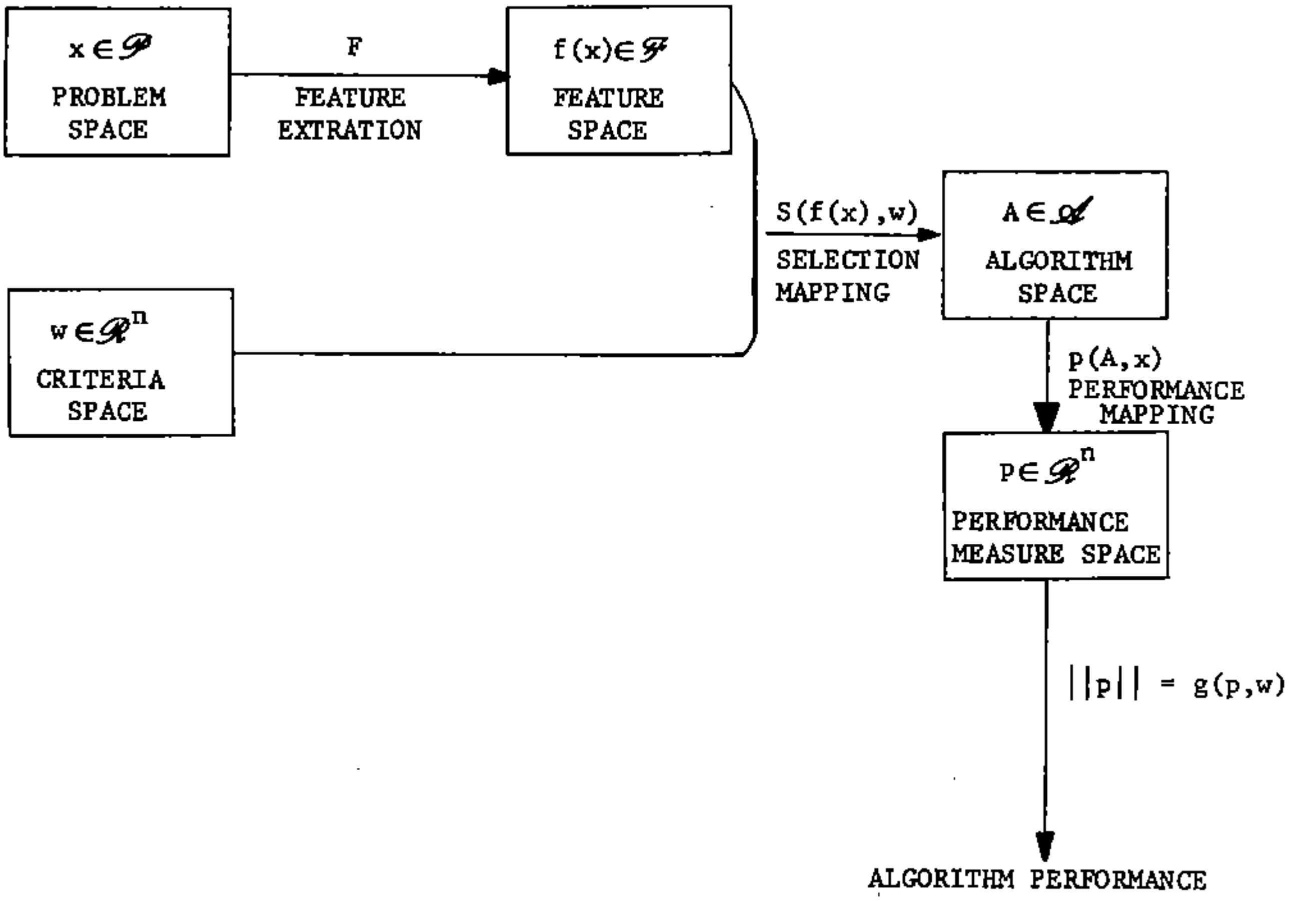

Figure 4. Schematic diagram of the model with selection based on problem features and variable performance criteria.

The additional definition for this model is:

g - Norm function from $\mathscr{R}^n$ x $\mathscr{R}^n$ to R^1 which measures the algorithm performance p(A,x) with the criteria w.

Some of the mappings now have changed domains, but their nature is the same. The choice of $\mathscr{R}^n$ for the criteria space is clearly arbitrary (and perhaps unnecessarily restrictive) but it is natural for the most common choice of the norm function: $g(p,w) - p \cdot w$.

We can at this point formulate new versions of the algorithm selection problem involving the criteria space. The variables of these formulations are:

Problem subclasses	$\mathscr{P}_0$
Algorithm subclasses	$\mathscr{A}_0$
Selection mapping subclasses	$\mathscr{S}_0$
Feature space	$\mathscr{F}$
Norm mapping	g

The number of interesting combinations is now quite large and we refrain from formulating all of them. Some of the more important problems are:

H. <u>Best Selection for a Given Criteria</u>. We assume that g(p,w) is known, that $\mathscr{F} = \mathscr{P}$ (and F is the identity) and w is given. The problem then is to determine that selection mapping B(x,w) which gives maximum performance:

$$g(p(B(x,w),x)w) \geq g(p(A,x),w) \qquad \text{for all } A \in \mathscr{A}$$

I. <u>Best Selection from a Subclass of Mappings for a Given Criteria and Feature Space</u>. We restrict S to a subclass $\mathscr{S}_0$ all mappings from $\mathscr{F}$ x $\mathscr{R}^n$ to A and, for a particular specified value of w and problem x, we wish to determine the best mapping $S^*(x,w)$ so that

$$g(p(S^*(f(x),w),x),w) \geq g(p(S(f(x),w),x),w) \qquad \text{for all } S \in \mathscr{S}_0 .$$

J. Best Selection from a Subclass of Mappings, Problems and Algorithms for a Given Criteria and Feature Space. This is a model of perhaps the most realistic situation. We have the feature space $\mathscr{F}$ and norm function g specified. We restrict ourselves to subclasses $\mathscr{S}_0$, $\mathscr{P}_0$ and $\mathscr{A}_0$ of selection mappings, problems and algorithms, respectively. Note we have $\mathscr{S}_0 : \mathscr{F} \times \mathscr{R}^n \to \mathscr{A}_0$. Within this framework we wish to select that mapping S^* so that

$$\max_{w \in \mathscr{R}^n} \quad \max_{x \in \mathscr{P}_0} \quad g(p(B(x,w),x),w) - g(p(S^*(f(x),w),x),w)$$

$$\leq \max_{w \in \mathscr{R}^n} \quad \max_{x \in \mathscr{P}_0} \quad g(p(B(x,w),x),w) - g(p(S(f(x),w),x),w)$$

for all $S \in \mathscr{S}_0$. Note that $g(p(B(x,w),x),w)$ is the best possible performance and the other g terms are the performances of the algorithms actually selected.

6. CONCLUSION. The abstract model presented in this report could be elaborated upon considerably. The study of the theoretical questions of the existence, uniqueness and characterization of best selection mappings and features mentioned in Section 2 can be expanded to fill a thick monograph. Those familiar with the mathematicians ability to develop theoretical structures from simple models can visualize how this would be done. However, the crucial point of a model is not its theoretical structure but its relevance to underlying real world problems. In other words, does this model allow us to develop better insight, understanding and analysis of real algorithm selection problems? This question is addressed in the next report.