Cross-Disciplinary Perspectives On Meta-Learning For Algorithm Selection

KATE A. SMITH-MILES
Deakin University, Australia

The Algorithm Selection Problem [Rice 1976] seeks to answer the question: *which algorithm is likely to perform best for my problem?* Recognising the problem as a learning task from the early 1990s, the machine learning community has developed the field of meta-learning, focused on learning about learning algorithm performance on classification problems. But there has been only limited generalisation of these ideas beyond classification, and many related attempts in other disciplines (such as AI and operations research) to tackle the algorithm selection problem in different ways, introducing different terminology, and overlooking the similarities of approaches. In this sense, there is much to be gained from a greater awareness of developments in meta-learning, and how the ideas can be generalised to learn about the behaviours of other (non-learning) algorithms. In this paper we present a unified framework for considering the Algorithm Selection Problem as a learning problem, and use this framework to tie together the cross-disciplinary developments in tackling the Algorithm Selection Problem. We discuss the generalisation of meta-learning concepts to algorithms focused on tasks including sorting, forecasting, constraint satisfaction, and optimization, and the extension of these ideas to bioinformatics, cryptography, and other fields.

Categories and Subject Descriptors: F1.3 [**Theory of Computation**]: Complexity measures and classes; I2.6 [**Artificial Intelligence**]: learning; I5.1 [**Pattern Recognition**]: models; F2 [**Analysis of algorithms and problem complexity**]
General Terms: algorithm selection, meta-learning
Additional Key Words and Phrases: classification, forecasting, constraint satisfaction, combinatorial optimization, sorting, dataset characterization, model selection, empirical hardness, landscape analysis.

---

1. INTRODUCTION

Researchers in a variety of disciplines have long sought a greater understanding of algorithm performance. Despite many empirical studies comparing various algorithms, much remains to be learned however about what makes a particular algorithm work well (or not) in a particular domain or a subset of classes. From the early work of John Rice [1976] which formalized the Algorithm Selection Problem, to the No Free Lunch (NFL) Theorems of Wolpert and Macready [1997] which stated that "...for any algorithm, any elevated performance over one class of problems is exactly paid for in performance over another class", it has long been acknowledged that there is no universally-best algorithm to tackle a broad problem domain. Instead, the NFL theorems suggest that we need to move away from a “black-box” approach to algorithm selection. We need to understand more about the characteristics of the problem in order to select the most appropriate algorithm that will ideally take into consideration known structural properties of the problem or instance. The key to characterising the problem is to devise suitable features that can be easily calculated, as acknowledged by Rice [1976] in the Algorithm Selection Problem. For a particular problem domain, what are the features or characteristics of particular instances of the problem that are likely to correlate with algorithm performance? Can we model the relationship between these characteristics and algorithm performance?

Author’s address: Kate Smith-Miles, School of Engineering and Information Technology, Deakin University, 221 Burwood Highway, Burwood, Victoria 3125, Australia.

To the machine learning community, the Algorithm Selection Problem is clearly a learning problem. It was natural for this community to focus their attention on the problem domain that interested them the most: learning algorithms applied to solve classification problems. The term meta-learning1, learning about learning, was first used in this context by Aha [1992], followed by the European project StatLog, whose goal was "to relate performance of algorithms to characteristics or measures of classification datasets". When presenting methods for determining which algorithm should be selected, the StatLog researchers [Michie et al. 1994] acknowledged the Algorithm Selection Problem of Rice [1976], but it is noteworthy that very little of the subsequent meta-learning literature draws this connection and commonality of goal. Instead we see divergence in the literature, with some researchers pursuing the ideas of Rice without knowledge of the meta-learning community's efforts, and the machine learning community focusing for the last 15 years or so on meta-learning without much connection to the formalism of Rice's Algorithm Selection Problem, or focus on the generalisation of their meta-learning ideas to other application domains beyond classification. The consequence is that the progress in the use of meta-learning concepts to solve the Algorithm Selection Problem in a variety of significant application domains has been slowed.

This paper aims to bring together the existing literature on both the Algorithm Selection Problem and meta-learning, presenting a unified framework for its generalisation across a broad range of disciplines including computer science, artificial intelligence, operations research, statistics, machine learning, bioinformatics, etc. The common factors in all of these disciplines, and the prerequisites for tackling the Algorithm Selection Problem using a (meta-) learning approach include:

i. the availability of large collections of problem instances of various complexities
ii. the existence of a large number of diverse algorithms for tackling problem instances
iii. performance metrics to evaluate algorithm performance
iv. the existence of suitable features to characterise the properties of the instances

Combining the features (iv) with the performance metrics (iii) across a large number of instances (i) solved by different algorithms (ii) creates a comprehensive set of meta-data or meta-knowledge about algorithm performance. Machine learning methods can be used to develop automated algorithm selection models, algorithm ranking models, combinations of algorithms, self-adaptive algorithms, etc. In addition, a knowledge discovery process can be adopted to yield greater theoretical insights into algorithm behaviour for problem instances of various complexities.

The remainder of this paper is organised as follows. Section 2 introduces the Algorithm Selection Problem as formalized by Rice [1976]. Learning about learning algorithm performance, or meta-learning, is presented in Section 3. The next section presents the literature available in the application of similar ideas to learning about regression, forecasting, sorting, constraint satisfaction, and optimization algorithm performance respectively. Section 5 then presents some of the possible extensions of these research directions: related meta-learning ideas beyond the Algorithm Selection Problem that could be adopted in these other application domains, as well as

[1] There are many definitions of meta-learning, all broadly seeking to exploit meta-knowledge about learning algorithm performance to improve the performance or selection of learning algorithms. The Algorithm Selection Problem relates to the 4th definition of meta-learning in the survey paper by Vilalta and Drissi [2002]: "Building meta-rules matching task properties with algorithm performance".

considering the generalisation of a learning-based approach to the Algorithm Selection Problem as it could be applied to new problem domains including bioinformatics, cryptography, and other emerging areas. A framework for achieving both automated algorithm selection and greater insights into improved algorithm design via an empirical learning approach of meta-data is also proposed. Finally, in Section 6, conclusions are drawn.

2. THE ALGORITHM SELECTION PROBLEM

The seminal paper by John Rice [1976] presented a formal abstract model that can be used to explore the question: With so many available algorithms, which one is likely to perform best on my problem and specific instance? The abstract model is reproduced with small modifications in Figure 1. There are four essential components of the model:

- the problem space *P* represents the set of instances of a problem class;
- the feature space *F* contains measurable characteristics of the instances generated by a computational feature extraction process applied to P;
- the algorithm space *A* is the set of all considered algorithms for tackling the problem;
- the performance space *Y* represents the mapping of each algorithm to a set of performance metrics.

The Algorithm Selection Problem can be formally stated as:

> For a given problem instance $x \in P$, with features $f(x) \in F$, find the selection mapping $S(f(x))$ into algorithm space A, such that the selected algorithm $\alpha \in A$ maximises the performance mapping $y(\alpha(x)) \in Y$.

Of course, the choice of features depends very much on the problem domain as well as the chosen algorithms. The features must be chosen so that the varying complexities of the problem instances are exposed, any known structural properties of the problems are captured, and any known advantages and limitations of the different algorithms are related to features. For example, when considering sorting algorithms for sequences of integers, the degree of pre-sortedness of the starting sequence is clearly a feature that will reveal the complexity of the instance, and will benefit some sorting algorithms more than others, depending on their mechanisms. Clearly, it is not straightforward to design suitable features for a given problem domain, and this is likely to have been a significant impediment to the widespread application of Rice's model.

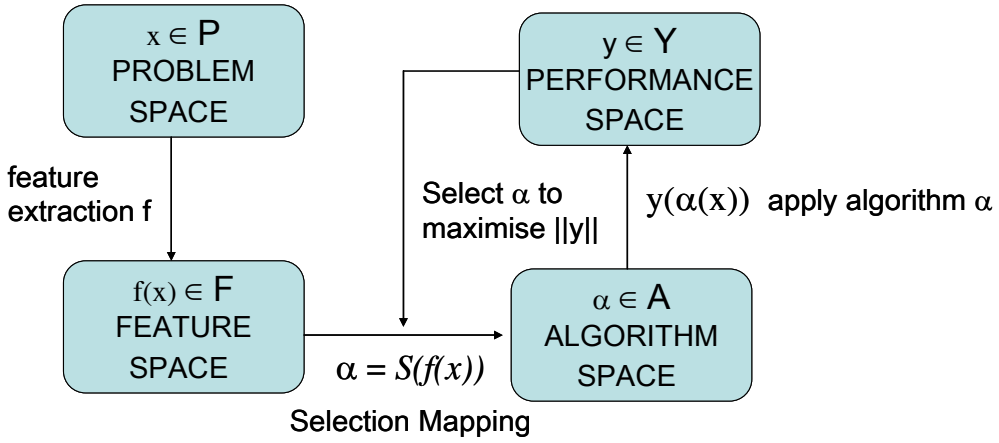

Figure 1: Schematic diagram of Rice's [1976] Algorithm Selection Problem model.
The objective is to determine the selection mapping S that returns the best algorithm α

Rice mentioned several application domains, including the selection of operating system schedulers and the selection of quadrature algorithms, but did not solve the problem. He mentioned the challenge in approximating the selection mapping function S and discussed several approximation methods including linear, piece-wise linear, and polynomial approximations. He concluded however, that "the determination of the proper non-linear form is still somewhat of an art and there is no algorithm for making the choice" [Rice 1976]. In fact the choice of the best algorithm for selecting the mapping function S is ironically another Algorithm Selection Problem. To the machine learning community though, this was a standard learning problem. Based on the training data ("meta-data") comprising input patterns *f(x)* and the output class showing which algorithm is best, the task is to learn the mapping S to produce a prediction of which algorithm will perform best for unseen test instances.

3. META-LEARNING FOR CLASSIFICATION PROBLEMS

While it is rare in the meta-learning literature to find reference to Rice's model of the Algorithm Selection Problem, it is useful to consider the notation and model he introduced as a common language to describe the various attempts by different communities to select the most suitable algorithm for different tasks. Casting meta-learning in the light of Rice's model, the machine learning community has been interested in classification problems (*P*), tackled using a variety of learning algorithms (*A*), with performance measures usually related to classification accuracy (*Y*). Through a succession of studies that will be discussed in this section, the features (*F*) used to characterise classification datasets have evolved in sophistication over the last 15 years, as have the learning algorithms used to learn the mapping *S*.

3.1 Foundations

Rendell and Cho [1990] presented one of the earliest attempts to characterise a classification problem and examine its impact on algorithm behaviour, using features related to the size and concentration of the classes. Aha [1992] extended this idea further to use rule-based learning algorithms to develop rules like:

If the given dataset has characteristics C1, C2, …, Cn

Then use algorithm A1 and don't use algorithm A2.

The algorithms considered for both the set A and to induce the rules (learn the mapping S) were IB1 (nearest neighbour classifier), CN2 (set covering rule learner), and C4 (decision tree). The problem instances were training datasets from the Frey and Slate [1991] letter recognition database (available from the UCI repository for machine learning databases, see Asuncion and Newman [2007]). The features included the number of instances, number of classes, the number of prototypes per class, the number of relevant and irrelevant attributes, and the distribution range of the instances and prototypes. An example of the kind of rules produced is:

IF (# training instances < 737) AND
(# prototypes per class > 5.5) AND
(# relevants > 8.5) AND
(# irrelevants < 5.5)
THEN IB1 will be better than CN2

One year later, Brodley [1993] extended the ideas to consider a more dynamic search for the best algorithm, developing rules to recognise when an algorithm is not performing well, and selecting a better one at run-time. When applied recursively, this dynamic form of algorithm selection can be used to create hybrid algorithms.

These early studies laid important foundations for the field of meta-learning. It became clear that an experimental approach could be taken to understanding the conditions under which algorithms performed well, where more analytical approaches were not suitable (see Rice's theorem [1956]). The potential extensions to these foundations included considering other algorithms beyond rule-based learners, a broader class of classification problems, more sophisticated features for dataset characterization, and other outcomes apart from selecting the best algorithm based on classification accuracy to include algorithm ranking or selection based on computation time, etc. The foundations for dynamic algorithm selection, and algorithm portfolios were also laid by this work. It is noteworthy that none of these early papers mentioned the Algorithm Selection Problem of Rice [1976], despite the commonality of goal.

3.2 First European Project StatLog (1991-1994)

StatLog: Comparative Testing of Statistical and Logical Learning, was an ESPRIT project concerned with the comparison of machine learning, neural and statistical approaches to classification. The principal aim of StatLog, as stated in the book that summarised the project results [Michie et al. 1994] was to provide an objective assessment of the strengths and weaknesses of the various approaches to classification. In addition, the analysis of these experimental results as presented in the chapter by Brazdil and Henery [1994], aimed "to relate performance of algorithms to characteristics or measures of classification datasets". The analysis extended the ideas of Aha [1992] to consider a broader range of classification datasets, learning algorithms, and features. The features measured various characteristics of each classification instance (training dataset) grouped according to simple measures (eg. size of datasets), statistical measures (eg. mean and variance of instance attributes) and information theoretical measures (eg. entropy) as presented in Table 1. The appeal of information theory measures [Shannon 1948] is that they can capture the randomness of an instance, which correlates with complexity, since an instance that appears to be quite random has few internal structures that can be exploited by an algorithm in finding a solution. Information theory measures, such as entropy and mutual information, can be calculated based on the degree of compressibility of an instance [Wallace and Boulton 1968].

Table 1: Features used to characterise classification datasets in the StatLog project

Measures	Definitions
Simple	
N	Number of examples
p	Number of attributes
q	Number of classes
Bin.att	Number of binary attributes
Cost	Cost matrix indicator
Statistical	
SD	Standard deviation ratio (geometric mean)
corr.abs	Mean absolute correlation of attributes
cancor1	First canonical correlation
fract1	Fraction separability due to cancor1
skewness	Skewness – mean of $\lvert E(X-\mu)^3\rvert / \sigma^3$
kurtosis	Kurtosis – mean of $\lvert E(X-\mu)^4\rvert / \sigma^4$
Information theory	
$H(C)$	Entropy (complexity) of class
$\overline{H}(X)$	Mean entropy (complexity) of attributes
$\overline{M}(C, X)$	Mean mutual information of class and attributes
EN.attr	Equivalent number of attributes $H(C) / \overline{M}(C, X)$
NS.ratio	Noise-signal ratio $(\overline{H}(X) - \overline{M}(C, X)) / \overline{M}(C, X)$

From the StatLog project, casting in the notation of Rice's Algorithm Selection Problem: P = 22 classification datasets from the UCI Repository; A = 23 machine learning, neural and statistical learning algorithms; Y = classification accuracy; and F = 16 features of the classification datasets as described in Table 1. The findings of the StatLog project confirmed that no single algorithm performed best for all problems, as supported by the NFL theorem. A decision tree algorithm (C4.5) was used to learn rules for each algorithm (the mapping S) describing the set of features and their values that made the algorithm the best performing one. This presented a binary outcome for each algorithm. For example, rules were generated for the CART (classification and regression tree) algorithm like

CART recommended IF (N <= 6435 AND skew > 0.57)

CART not recommended IF (N > 6435 OR (N < 6435 AND skew < 0.57)).

The StatLog project provided valuable meta-data that was widely used by meta-learning researchers for many years. Gama and Brazdil [1995] extended the results to use regression, rather than a decision tree, to learn the mapping S, producing estimates of classifier error. Lindner and Studer [1999] learned the mapping S using case based reasoning, also extending the range of features to include statistics related to discrimination power, statistical distribution tests, and distribution of outliers. Another significant direction following StatLog was the NOEMON approach [Kalousis and Theoharis 1999] where pairs of algorithms are used to create meta-learning models that recommend one algorithm over the other (or a tie) based on tests of statistical significance. Each meta-learner customises the features used to ensure relevance for the pair of algorithms, and the system provides a recommendation by combining the outputs of the individual meta-learners.

StatLog also introduced the idea that sometimes the effort in calculating features is greater than running simple algorithms. It posed the question of whether we could learn to predict the performance of different algorithms with reference only to the performance of simple (yardstick) algorithms. This concept later became known as landmarking [Pfahringer et al. 2000], and became another key direction in meta-learning research. The StatLog project acknowledged the Algorithm Selection Problem of Rice [1976] and the importance of generating features that allow the strengths and weaknesses of the different algorithms to be exposed. After a few years of working with the limited meta-data however, researchers were keen to extend the meta-data and approaches for learning the mapping S.

3.3 Second European Project: METAL(1998-2001)

Based on the success of StatLog and the progress of meta-learning research, a second ESPRIT project was funded in 1998 titled METAL: A meta-learning assistant for providing user support in machine learning and data mining. Focusing on classification and regression problems, METAL aimed to develop model selection and method combination approaches provided to users with the support of an online environment. Before these approaches were developed though, new meta-data was created including more classification datasets, and a more focused set of learning algorithms with computational performance results based on 10-fold cross-validation.

The meta-data for the METAL project, in the notation of Rice's Algorithm Selection Problem, was: P = 53 classification datasets from UCI repository and other sources; A = 10 learning algorithms (3 decision tree classifiers, 2 rule-based classifiers, 2 neural networks, instance-based learner IB1, naïve bayes, and linear

discriminant) ; Y = accuracy and time performance based on 10-fold cross-validation; F = 16 features as created by the StatLog project. The METAL dataset now contains additional datasets.

Algorithm ranking, rather than the binary focus of the StatLog project, became the basis for the decision support tool, utilising the work of Brazdil et al. [2003]. An instance-based learning algorithm (K-nearest neighbour) was used to determine which training datasets are closest to a test dataset based on similarity of features, and then predict the ranking of each algorithm based on the performance of the neighbouring datasets. In addition to accuracy measures, time performance of algorithms was considered simultaneously in a multi-criteria measure.

A number of important studies were conducted as part of the METAL project, including extending the range of features to include the structural characteristics of decision tree models for each dataset (such as the number of nodes and leaves, width and depth of the tree, distribution of attributes and nodes in the tree, etc.) and investigating the optimal set of features to include in the model [Bensusan et al. 2000; Peng et al. 2002]. The idea of landmarking as an alternative to generating computationally expensive features was also developed further during METAL to create models predicting the performance of algorithms relative to the performance of simpler and faster learners [Pfahringer et al. 2000; Furnkranz and Petrak 2001; Soares et al. 2001]. Alternative ranking approaches have also been investigated, as well as methods for combining meta-classifiers [Berrer et al. 2000; Kalousis and Hilario 2001; Seewald et al. 2001; Bensusan and Kalousis 2001; Todorovski et al. 2002; Hilario 2002; Todorovski and Dzeroski 2003; etc.].

3.4 Other Recent Developments

Lim, Loh, Shih [2000] extended the meta-data from the StatLog project to consider the performance of A = 33 algorithms (22 decision tree, 9 statistical and 2 neural network classifiers) on a collection of P = 32 datasets, half of which were generated by adding noise to existing datasets. In addition to classification accuracy and the size of the trees generated, they also recorded computational time and the scalability of the classifiers as the sample size increases as performance metrics Y. There seems to have been very little use of this data so far however in a meta-learning context as we have been discussing.

A series of studies have been conducted on new meta-data, which extended the number of classification datasets and considered alternative approaches to learning the mapping S. Smith et al. [2001] generated P = 57 classification datasets from the UCI Repository, A = 6 learning algorithms (3 rule-based and 3 statistical); Y = rank based on accuracy; F = 21 features based primarily on the StatLog measures. A supervised neural network was used to learn the mapping S, although the size of the training set (meta-data) was noted as being rather limited for this kind of process, with a danger of overfitting. Ten-fold cross-validation was used to generate test set performance results. When predicting the best performing algorithm, 77% accuracy was obtained, increasing to 95% and then 98% when inspecting the top 2 and top 3 predicted algorithms for the winner.

Concerns over the validity of using supervised learning methods for learning the mapping S on limited meta-data resulted in an unsupervised approach being used on the same meta-data. Smith et al. [2002] used a self-organising feature map (see Kohonen [1988]) to cluster the 57 classification datasets based only on the features, producing 5 clusters of datasets. Each cluster was then inspected to identify common features and to evaluate the performance of different algorithms within each cluster. A series of rules were inferred from a statistical analysis of the clusters suggesting to which characteristics of the datasets certain algorithms should be suited.

The largest collection of meta-data for classification thus far seems to be the study of Ali and Smith [2006]. Adopting the notation of Rice's Algorithm Selection Problem, this study generated meta-data with: P = 112 classification datasets from the UCI Repository; A = 8 learning algorithms (3 rule-based, 1 neural, and 4 statistical) including Support Vector Machines (SVM) for the first time; Y = rank based on an F-measure that combines a weighting of cost-sensitive accuracy as well as computational time; F = 31 features derived from the StatLog characteristics, with the addition of several new statistical features from the Matlab toolbox and other sources [Gnanadesikan 1997] such as index of dispersion, centre of gravity, and maximum and minimum eigenvalues of the covariance matrix. A decision tree C4.5 was used to learn the mapping S, producing rules for each algorithm with average accuracy of ten-fold cross-validation testing exceeding 80% accuracy in predicting the best algorithm.

3.5 Meta-learning for classification beyond algorithm selection

Beyond the task of algorithm selection there are many ways in which the ideas presented in the above sections can be generalised to related tasks. For example, within a particular algorithm there is often the need to select optimal parameter settings, and an experimentally driven meta-learning approach can be adopted here too. Ali and Smith-Miles [2006] generalised the approach to learn the best kernel to use within support vector machines (SVM) for classification. Using Rice's notation: P = 112 classification datasets from UCI repository; A = 5 kernels within SVM (polynomial, radial basis function, Laplace, spline, multiquadratic); Y = classification accuracy; F = 29 features, similar to StatLog but extended to include 11 statistical, 3 distance-based, 15 density based measures. Again, rules were generated to learn the mapping S using C4.5, producing accuracies exceeding 80% on ten-fold cross-validation testing. Of course, selecting a different kernel for SVM essentially changes the algorithm, so this too can arguably be considered the same task as algorithm selection. It establishes though a generalisation of concept that can be extended. A similar approach was adopted to determine the best method for selecting the width of the RBF kernel, using the same meta-data [Ali and Smith 2005]. A framework for using meta-learning to optimise parameter selection is presented in Duch and Grudzinski [2001].

There are also many other types of meta-learning that have been developed over the last 15 years or so, as summarised in the survey article by Vilalta & Drissi [2002]. The preceding sections have been focused only on their 4th definition of meta-learning: "Building meta-rules matching task properties with algorithm performance". However, the meta-learning community has also developed many other approaches to learning about learning algorithm performance of relevance to classification problems. These include dynamically adjusting the inherent bias in a learning model [DesJardins and Gordon 1995; Castiello et al. 2005]; combining models via stacked generalisation [Wolpert 1992; Chan and Stolfo 1997], ensembles [Opitz and Maclin 1999] and other approaches [Gama and Brazdil 2000; Todorovski and Dzeroski 2003; Peterson and Martinez 2005]; meta-learning of broader data mining processes that include algorithm selection and other tasks [Prodromidis et al. 2000; Bernstein et al. 2005]; and landmarking instead of using features for dataset characterization [Pfahringer et al. 2000; Ler et al. 2005].

All of these tasks are called meta-learning because of their focus on learning about learning algorithms. But the ideas presented can be readily extended to other types of algorithms and problems, as will be discussed later in this paper. For now, we return to the Algorithm Selection Problem of Rice [1976]. It should be noted that the terminology evolved by the machine learning community began to refer to model selection rather than algorithm

selection, although these are basically the same concepts as recognised by Rice. There is very little reference to Rice's Algorithm Selection Problem in the meta-learning literature after the StatLog project [Michie et al. 1994]. In some other communities however, such as artificial intelligence, Rice [1976] has been the starting point for developments in algorithm selection, with very little reference to the meta-learning literature that was evolving at the same time. In the following sections we examine how meta-learning ideas have been extended beyond classification problems to assist with algorithm selection in other domains, and what further potential exists.

4. GENERALISATIONS TO OTHER DOMAINS

A learning approach to the Algorithm Selection Problem can be applied to any problem domain provided there is sufficient data available to learn a mapping S from the meta-data $<P, A, Y, F>$. The greatest challenge is the derivation of suitable metrics as features to characterize the datasets. In this section we will discuss the limited existing generalisations of a learning approach to Algorithm Selection using this framework in a number of cross-disciplinary domains.

4.1 Regression

One of the objectives of the METAL project was to extend the approach to consider regression problems in addition to classification problems. As soon as we consider a new problem domain however, the most important question becomes *what are the suitable features that can adequately characterise a dataset?* While the StatLog features are suitable for classification problems (eg. measuring the distributions between classes, entropy of classes, etc) they need to be adapted for regression problems that model a relationship between inputs and a continuous single output or target variable. Kopf et al. [2000] tested the suitability of meta-learning applied to regression problems using the StatLog features primarily. Using the notation of Rice (1976), they considered: P = 5450 instances of regression problems with sample sizes of each instance (training data) in the range (110,2000) and the number of variables per instance in the range (2,10); A = 3 regression models (LDA, QDA and a neural network); Y = error rate as measured by MAD, MSE, NMAD and NMSE; F = 25 features based primarily on the StatLog measures with some additional ones relating to tests of normality and Box's M statistic. The mapping S was learned using the decision tree algorithm C5.0.

Kuba et al. [2002] then developed new features for regression problems. These included coefficient of variation, sparcity, and stationarity of the target variable; presence of outliers in the target; R^2 coefficient of a linear regression model; and average absolute correlations between the variables themselves, and between the variables and target. These measures were in addition to the suitable StatLog measures as tested by Kopf et al. [2000]. The features selected by Kuba et al. [2002] were used in the study by Soares et al. [2004] which investigated a meta-learning approach to selecting the width of the Gaussian kernel parameter in support vector machine regression. Using the notation of Rice [1976] they studied: P = 42 regression instances; A = 11 candidate values for the width parameter α; Y = error as measured by NMSE; F = 14 features as presented by Kuba et al. [2002]. The mapping S was learned using the algorithm ranking method (based on the k-nearest neighbour algorithm) presented in Brazdil et al. [2003]. The study by Soares et al. [2004] was really focused on parameter selection for a particular regression method (SVM), rather than algorithm selection for regression problems though, and there have been relatively few studies published on the latter.

4.2 Time-Series Forecasting

There are several early works adopting a learning approach to time series forecasting model or algorithm selection. Arinze [1995] extended the previous expert systems approaches to deriving rules for forecasting method selection to propose a machine learning approach. While he did not make reference to Rice [1976] or the efforts that were underway with StatLog, it is the same general approach that was proposed. Arinze et al. [1997] demonstrated the approach with empirical results. Casting their work in the framework of Rice [1976]: P = 67 time series datasets; A = 3 forecasting methods: adaptive filtering (AD), Holt's exponential smoothing (HT), and Winter's method (WT); they were also combined with 3 hybrid methods: moving average + HT, AD+WT, HT+WT; Y = average standard error; F = 6 features proposed in Arinze [1995] which were granularity of data (quarterly or yearly data), number of turning points, autocorrelation coefficient, trend using regression coefficient, coefficient of determination from regression model, mean squared error from regression model. The mapping S was learned using a knowledge-based system (1st class) to generate rule-based induction.

Arinze's studies were followed by Venkatachalam and Sohl [1999] who proposed a two-stage neural network approach for learning the mapping S. In the first stage a time series is mapped, based on features, to one of three groups of algorithms. These groups are i) flexible algorithms adaptive to a variety of trends (Holt 2 Parameter Linear Exponential Smoothing (ES), Winter 3 Parameter ES, Brown Triple ES); ii) algorithms responsive to linear trends (Brown Double ES, linear regression, Adaptive Response ES), and iii) simple algorithms (Naïve deseasonalized, Single ES, Simple Moving Average). Thus there are 9 algorithms considered, grouped into three groups. Once the first neural network has determined which group is most suitable for a given time series, a second neural network is then applied to determine which of the three algorithms within the group is likely to produce the smallest forecasting error. Using Rice's notation: P = 180 time series from the M3 Competition [Makridakis and Hibon 2000]; A = 9 algorithms divided into three groups as described above; Y = mean absolute percentage error (MAPE); F = a set of six features measuring the length of the time series, time period, type of data (eg. macro, micro, demographic, etc.), basic trend, recent trend, and variability of the series as measured by R2 of a regression model. Venkatachalam and Sohl [1999] demonstrated an accuracy of predicting the most accurate forecasting algorithm of around 50%.

More recently, Prudêncio and Ludermir [2004] presented two case studies of increasing sophistication. For their first case study: P = 99 stationary time series from the Time Series Data Library (see Hyndman [2002]); A = 2 forecasting algorithms (simple exponential smoothing and time-delay neural network); Y = mean absolute error; F = 14 features measuring the length of the time series; statistical properties of autocorrelations; coefficient of variation to ascertain stability; skewness and kurtosis; and tests of turning points as a measure of randomness. The mapping S was learned using J4.8 (a Java implementation of the C4.5 decision tree algorithm in Weka, see Witten and Frank [2005]) to determine rules for deciding which of the two algorithms is expected to produce the smallest error.

For their second case study, Prudêncio and Ludermir [2004] extended the work to consider ranking of algorithms using the NOEMON approach [Kalousis and Theoharis 1999]. They also extended the meta-data: P = 645 yearly time series of the M3-Competition [Makridakis and Hibon 2000]; A = 3 common forecasting algorithms used in the M3-Competition: random walk, Holt's linear exponential smoothing, and auto-regressive model; Y = ranking based on error; F = a refined subset of the features from the first case study, plus a measure of trend, and a categorical variable indicating the source of the time series (eg. finance, demographic, etc.). The

mapping S was learned using a combination of pairwise-trained neural network classifiers following the NOEMON approach. An accuracy of around 75% was obtained in selecting the best algorithm.

Simultaneously, Wang [2005] had studied a different set of features that could be used to characterize time-series datasets, and their use in a meta-learning context. These features were also used to generate clusters of time series, and compared well to existing time series clustering approaches due to their ability to capture the global characteristics of the times series [Wang et al. 2006]. For these studies, adopting the notation of Rice [1976]: P = 315 time series datasets from UCR Time Series Data Mining Archive (see Keogh and Folias [2002]), the Time Series Data Library (see Hyndman [2002]) and several synthetic datasets with defined characteristics relating to trend, seasonality, noise, etc.; A = 4 forecasting methods: exponential smoothing, ARIMA, random walk, and a backpropagation neural network; Y = relative absolute error; F = 13 features (some based on raw data and some on time series adjusted for trend and seasonality) characterizing trend, seasonality, serial correlation, non-linearity, skewness, kurtosis, self-similarity, chaos, and periodicity. In order to learn the mapping S two approaches were adopted. Firstly, rules were learned using C4.5 following the meta-learning community. Secondly, the time series were clustered based on the 13 features, and rules were inferred based on cluster membership and examined performance of the different algorithms within each cluster. A variety of clustering approaches were considered, including self-organising feature maps [Kohonen 1988] and hierarchical clustering. Accuracies of over 80% were obtained for predicting which algorithm should be selected. The usefulness of the meta-data for clustering problems was also illustrated [Wang et al. 2006] and suggests an interesting approach for unsupervised learning of meta-data for knowledge discovery not often considered by the meta-learning community.

4.3 Sorting

With numerous sorting algorithms available (see for example Estivill-Castro and Wood [1992]) it is not surprising that the domain of sorting has been tackled using the framework of Rice's [1976] Algorithm Selection Problem. Lagoudakis et al. [2001] expressed the task of sorting algorithm selection by considering the length of the sequence of integers to be sorted as the single feature, and using dynamic programming to estimate the mapping S via a Markov Decision Process. Rules, or optimal policies, were generated showing which of the three considered algorithms (InsertionSort, MergeSort or QuickSort) should be used depending on the length of the sequence to be sorted. It was a small study, but the first to link sorting algorithm selection to the framework of Rice [1976], and show its potential.

A larger study of sorting algorithms was conducted by Guo [2003] who extended Lagoudakis et al.'s approach to consider more sorting algorithms, additional features to characterise a sequence, and a learning approach for the mapping S. Guo recognised that a sequence of integers to be sorted could be characterised by more than its length, but also by its degree of presortedness. He utilised three of the eleven measures of presortedness presented in Estivill-Castro and Wood [1992]: the number of inversions (INV), the number of runs of ascending sub-sequences (RUN), and the length of the longest ascending sub-sequence (LAS). Using a Markov process, four classes of random sequences were generated of varying size and degrees of presortedness. Using the notation of Rice [1976], Guo's [2003] study can be summarised as: P = 43195 instances of random sequences of different sizes and complexities; A = 5 sorting algorithms (InsertionSort, ShellSort, HeapSort, MergeSort, QuickSort); Y = algorithm rank based on CPU time to achieve sorted sequence; F = 3 measures of

presortedness (INV, RUN, LAS) and length of sequence (size). The mapping S was learned from this meta-data using three different machine learning methods: C4.5 to produce rules, and two classifiers (Naïve Bayes and a Bayesian network learner). A dynamic approach to algorithm selection was also adopted for the recursive sorting algorithms, following the ideas of Brodley [1993]. Over 90% accuracy was achieved in selecting the fastest sorting algorithm.

Interestingly, Guo [2003] makes reference to Rice [1976] and Brodley [1993], and even refers to the method used to learn the mapping S as a "machine learning-based" approach, but there is no reference to the term "meta-learning" or any of the seminal papers in the meta-learning community by Brazdil and colleagues. Here we see a clear divergence in the literature, and one that will become even more pronounced when we consider the work in the constraint satisfaction and optimization domains.

4.4 Constraint Satisfaction

For many decades the constraint programming and artificial intelligence communities have focused on algorithms for solving NP-hard constraint satisfaction problems such as the SAT problem: a generic constraint satisfaction problem with binary variables and arbitrary constraints expressed as clauses. In recent years considerable progress has been made in studying the *empirical hardness* of various instances of such problems. A famous result by Selman et al. [1996] demonstrated the existence of algorithm independent phase transitions for random 3-SAT problems, showing that when the ratio of the number of clauses to variables is approximately 4.26 the instance becomes hard (ie. takes longer to solve for all algorithms). Following these findings, much research has been conducted in the artificial intelligence community to study algorithm performance empirically, particularly focusing on identifying features that correlate with the empirical hardness of problem instances (see for example Hoos and Stützle [1999]; Slaney and Walsh [2001]; Boukeas et al. [2004]; Achlioptas et al. [2005]).

In 2001, Horvitz et al. proposed a Bayesian approach to learning the mapping S between the features (that they called structural evidence) and CPU time (that they called execution evidence) of two solvers for tackling the Quasigroup with Holes (QWH) problem, an NP-complete problem based on Latin Squares with applications to timetabling, routing and statistical experimental design. Rather than tackling the problem as an algorithm selection problem though, to automatically select which of the two algorithms (CSP-ILOG and Satz-Rand solvers) should be used to obtain the fastest solution, the Bayesian approach was used to focus on one algorithm at a time and seek explanations of features that predict runtime. Adopting the notation of Rice [1976] we can summarise their work as: P = 5000 QWH instances of size 34 with 380 unassigned variables; A = CSP-ILOG and Satz-Rand algorithms analysed separately; Y = CPU time to find a solution; F = unnamed features of the problem instances (specific to Latin Squares), statistics recorded by Satz-Rand of a dynamic solution (eg. number of current unbounded variables), and statistics recorded by the CSP-ILOG solver. The aims of the proposed approach were to tackle the problem of predicting runtime in real-time (a dynamic approach), and hence the use of features relating to the performance of the selected algorithms during run-time. This is one of the earliest approaches to dynamic algorithm selection where the aim was "to learn predictive models to refine and control computational procedures as well as to gain insights about problem structures and hardness" [Horvitz et al. 2001].

In 2002 the idea of learning, in a more static fashion, the relationship between the features of an instance and the time taken by an algorithm to solve constraint satisfaction problems was furthered by Leyton-Brown et al. [2002]. Technically the problem tackled by Leyton-Brown et al. [2002] is more than a constraint satisfaction problem, in fact the winner determination problem they tackled is a constrained optimization problem. We mention it here however since it laid the foundation for their future work on constraint satisfaction. Actually, this work was not an attempt at algorithm selection either, since only one algorithm was studied empirically, but there is much similarity of concept to the formalism of Rice [1976]. Although no reference was made to the Algorithm Selection Problem, or any of the existing meta-learning literature, we can cast this work in the framework of Rice as follows: P = 8544 instances of the combinatorial auction winner determination problem, generated randomly across three different instance sizes; A = a single algorithm CPLEX 7.1; Y = computation time to solution of the problem; F = 25 features relating to statistical characteristics of the graph representations of the instance (degree statistics, edge density, average minimum path length, etc.), features derived from solving a linear programming relaxation of the problem (norms of the integer slack variables), and statistics of the prices in the auction problem. The mapping S was learned using a linear regression model, as well as 2nd degree polynomial and spline models, to predict the logarithm of CPLEX CPU time.

Throughout a series of papers over several years, Leyton-Brown and co-authors extended the methodology to consider algorithm selection in algorithm portfolios (see Leyton-Brown et al. [2003a]), and turned their attention to the SAT problem (see Nudelman et al. [2004]). Their work on algorithm portfolios, a term that has become popular in the artificial intelligence community more so than the machine learning community (see for example, Gagliolo and Schmidhuber [2006]), acknowledged the relationship to Rice [1976], and can be summarised as: P = 4500 instances of the winner determination problem; A = 3 algorithms (CPLEX, Gonen-Lehmann and CASS, both based on branch and bound); Y = computation time to solve the instance; F = same set of 25 features used in Leyton-Brown et al. [2002]. The mapping S was learned using linear regression to predict the logarithm of runtime.

Their work on using a learning approach to understand the relationship between the features of random SAT problems and the computation time of various solvers (see Nudelman et al. [2004]) can be summarised as: P = 20000 uniformly-random 3-SAT instances with 400 variables each, and the clauses to variables ratio uniformly selected from the range (3.26, 5.26); A = 3 algorithms (kcnfs, oksolver, satz) which performed well on SAT competitions; Y = computation time to reach a solution; F = a set of 91 features characterising a SAT instance based on problem size features (number of clauses and variables, and several ratios), statistical features of the variable-clause graph, variable graph, and clause graph, ratios of positive and negative literals in each clause, features based on LP relaxation (objective value, fraction of binary variables, variable integer slack statistics), search space size estimates based on DPLL algorithm statistics, and several measures based on local search statistics using 2 algorithms, GSAT and SAPS. The mapping S was again learned using a linear regression model to predict CPU time. Their comprehensive set of features essentially combines features that can be calculated immediately based on the instance data, as well as more computationally expensive features that require other algorithms to be run first and statistics collected based on their performance. Although it is never mentioned in their work, this latter set of features is related to the idea of landmarking developed by the machine learning community. The work of Nudelman et al. [2004], as well as Horvitz et al. [2001], can thus be seen as generating a feature set that combines problem specific characteristics with more generic landmarking-type features.

In Xu et al. [2007a] these ideas of algorithm portfolio design are extended further, and the authors' prize winning approach, SATzilla-07, for the 2007 SAT competition is described. Their algorithm portfolio is essentially an algorithm selection tool, trained on a number of empirical hardness models. Re-casting using Rice's notation, the training of the portfolio for their 2007 competition entry consisted of: P = 4811 SAT instances from previous SAT competitions and 2006 SAT race, comprising a mix of random, handmade (crafted) and industrial instances; A = 7 high performance solvers (Satzilla-07, Eureka, Zchaff_Rand, Kcnfs2006, Minisat2.0, March_dl2004, Vallst and Rsat), plus those same 7 solvers with preprocessing (see Xu et al. [2007] for details); Y = log of CPU time; F = 48 features from Nudelman et al. [2004]. The mapping S was learned using a ridge regression method to predict log CPU time, creating a set of empirical hardness models for each solver.

Xu et al. [2007b] have also examined the interesting problem of whether a constraint satisfaction problem is satisfiable or not. Treating the learning as a classification task, they used similar meta-data to their previous studies, modelling the satisfiability using sparse multinomial logistic regression. This is an interesting variation on the theme of algorithm selection.

Rather than focusing on the selection of the best algorithm from a pre-defined set of algorithms in a portfolio, Samulowitz and Memisevic [2007] have recently proposed on a more dynamic approach that selects components of algorithms in real-time. Using the QBF problem (a type of SAT problem) as a vehicle for illustrating the approach, they focus on dynamically predicting the optimal branching heuristic in search-based QBF solvers. For their study, P = 897 QBF instances from QBFLib 2005 and 2006, plus 800 additional random instances; A = 10 branching heuristics operating with the DPLL solver; Y = run time; F = 78 features of QBF instances relating to the properties of the variables, clauses, quantifier alternations, and literal appearances. The mapping S was learned using multinomial logistic regression. This study relates more to the growing body of work on dynamic algorithm selection [Armstrong et al. 2006; Streeter et al. 2007] and tuning of parameters via racing algorithms [Maron and Moore 1997], rather than the focus of this paper on the more static algorithm selection problem. Nevertheless, the generalisation of the concepts is clear.

4.5 Optimization

Typical optimization problems involve finding a solution to a problem that minimises the cost (or maximises the benefits) while simultaneously satisfying some constraints. Thus optimization problems can be seen as an extension of the constraint satisfaction paradigm to include a balance of two conflicting goals. It is often easy to find an expensive solution that satisfies the constraints, or to find a good solution that violates some constraints, but challenging to satisfy both of these goals simultaneously. A large class of optimization problems are known as combinatorial optimization problems (COPs) which involve binary decision variables. Well-known examples of COPs include the famous Travelling Salesman Problem (TSP) whose task is to find the shortest possible path through a set of cities (minimal cost) while ensuring that all cities are visited exactly once (constraint satisfaction). The binary decision variables tell us which cities are connected in the salesman's solution.

There are two broad approaches to solving such problems. The first solves the problem exactly, but may not reach a solution for very large problems due to computational and memory limitations. Branch-and-bound algorithms are common in this approach. The question to ask here is how much run time is needed until the best solution is found? Where the problem is solvable exactly, the algorithm selection problem can focus on selecting

the best exact algorithm, where best is defined according to fastest run time. The second approach, if the problem is not solvable exactly, is to find a faster near-optimal solution using heuristics or stochastic search algorithms, focusing on the questions: how much run time is needed until a satisfactory (not necessarily optimal) solution is found, or how good is the solution for a fixed run time? The algorithm selection problem focuses here on selecting amongst heuristics, where the performance criterion is either solution time or quality. As discussed in the previous section, the constraint programming community has been seeking greater understanding of the relationship between instance characteristics and the run-time performance of branch-and-bound types of solvers and other constraint satisfaction algorithms. With the additional complexity of a cost function that needs to be minimised, however, the operations research community has focused in recent years on stochastic search algorithms and meta-heuristics such as simulated annealing, tabu search, ant colony optimisation, genetic algorithms, etc. (see for example the various chapters in Glover and Kochenberger [2003]) to achieve solutions to complex optimization problems within a reasonable computation time.

There have recently been calls from the meta-heuristics community for developing greater insights into algorithm performance by studying search space or problem instance characteristics. According to Stützle and Fernandes [2004], "currently there is still a strong lack of … understanding of how exactly the relative performance of different metaheuristics depends on instance characteristics". A number of researchers have been focusing in recent years on landscape analysis since "gathering knowledge about a landscape could help to refine existing meta-heuristics well adapted for certain types of instances or problems, or give ideas for developing new methods, enhancing the number of tractable instances" [Angel and Zissimopoulos 2002]. Certainly, these directions give due consideration to the No Free Lunch Theorem of Wolpert and Macready [1997], and acknowledge that the only solution is to move away from a "black-box" approach to algorithm selection, and to develop greater understanding of the characteristics of the problem in order to select the most appropriate algorithm. This is precisely the argument of Rice in 1976, also rarely cited by the meta-heuristics community.

For many NP-hard optimisation problems there is a great deal we can discover about problem structure. Landscape analysis (see Knowles and Corne [2002]; Merz [2004]) is one framework for measuring the characteristics of problems and instances, and there have been many relevant developments in this direction, but the dependence of algorithm performance on these measures is yet to be completely determined. In recent years a number of studies have been conducted, particularly focused on the Travelling Salesman Problem and the Quadratic Assignment Problem (QAP) to explore the complexity and hardness of particular instances of these problems. Like the SAT problem, it has recently been demonstrated [Achlioptas et al. 2005] that these problems undergo a phase transition of sorts, whereby the problem can be considered easy or hard depending on the characteristics of the data (eg. the location of TSP cities in the plane). These phase transitions create different landscape complexities, and different algorithms cope with these complexities in different ways (see Angel and Zissimopoulos [2002]; Merz [2004]; van Hemert [2006]). Other relevant measures for characterizing the complexity of an optimization problem, and its associated landscape include Kolmogorov complexity [Borenstein & Poli 2006], the size of basins of attraction [Merz 2004]; intrinsic instance hardness [Boukeas et al. 2004], landscape ruggedness coefficients [Angel & Zissimopoulos 2002], entropy and diameter of a population of solutions [Bachelet 1999], fitness distance correlation [Jones & Forrest 1995], and flow dominance for QAP problems [Vollmann & Buffa 1966], in addition to measures of problem size (the number of variables and constraints), statistical properties of instances, measures of sparsity, etc.

Despite the fact that the metaheuristics community seeks greater insight into algorithm performance by studying search space and instance characteristics, and the existence of many relevant metrics for characterizing the complexity of an optimization problem instances, there has been surprisingly little attempt to generalize the relevant meta-learning ideas developed by the machine learning community, or even to follow some of the directions of Leyton-Brown et al. in the constraint programming community. The metaheuristics community is now well-placed to follow such directions. There has been recent work on developing benchmark instances of COPs like the QAP that intentionally display a wide range of complexities [Stützle and Fernandes 2004]; availability of a wide variety of algorithms of diverse behaviours; and many suitable measures of instance complexity and hardness. All that remains to be done is to learn the relationships between the instance characteristics and algorithm performance to assist with algorithm selection.

There have been some studies that have come close to achieving this, but their objectives were different. Telesis and Stamatopoulos [2001] have studied how algorithm performance varies with hardness of knapsack and set partitioning problems using a small set of features relating to the average values of vectors defining the problem. They used kernel regression to model the relationship between the characteristics of the problem at a given iteration, and the likely objective function for a given heuristic to guide the search from that point forward. Thus their machine learning approach is focused on dynamic search, rather than static algorithm selection. Another direction, somewhat related to generalization of meta-learning ideas to optimization lies in the hyper-heuristic approach developed by Burke and co-authors (see Burke et al. [2003] for a review). Hyper-heuristics seek to develop an intelligent heuristic approach to choosing the best heuristic for solving an optimization problem. Rather than measuring complex characteristics of the search space however, a number of simple (low-level) heuristics are used to characterize the complexity of the instance, and a weighted linear combination of their performance metrics is used to determine the best low-level heuristic to call at any time during a dynamic search. The similarities to the landmarking concept developed by the meta-learning community are quite strong.

Smith-Miles [2007] has recently proposed the use of meta-learning ideas to model the relationship between the instance characteristics and algorithm performance for the QAP. Expressed using the notation of Rice [1976] she used: P = 28 instances of the QAP taken from Stützle and Fernandes [2004]; A = 3 metaheuristics: robust tabu search, iterated local search, min-max ant system; Y = % difference between objective function of heuristic versus known optimal solution (performance gap); F = 4 measures of problem size (dimensionality, dominance of distance and flow matrices, sparcity of matrices), and 4 measures based on iterated local search runs (number of pseudo-optimal solutions, average distance of local optima to closest global optima, fitness distance correlation coefficient, etc.). A number of different experiments were conducted to generate mappings S. A neural network was used for predicting the performance gap to be expected by different metaheuristic algorithms, as well as the ranking of algorithms. The study also used self-organising maps to select the best algorithm by creating visual explorations of the performance of different algorithms under various conditions describing the complexity of the problem instances. While this is only a preliminary study of limited size and scope, it has nonetheless demonstrated the relevance of meta-learning ideas to the optimization domain. It is a promising direction that will surely assist the meta-heuristics community in their quest for greater understanding into "exactly how the relative performance of different metaheuristics depends on instance characteristics" [Stützle and Fernandes 2004].

This paper thus far has examined the existing work related to the algorithm selection problem of Rice [1976] and developments by a number of different research communities: the meta-learning approach of the machine learning community, the decision support systems approach of the forecasting community, the empirical hardness approach of the artificial intelligence and constraint programming communities, and the search space and landscape analysis approaches of the meta-heuristics community. These previous studies have been re-cast in the light of the common framework of Rice, as a first step to achieving greater linking of cross-disciplinary strands. A summary of the studies discussed is presented in Table 2.

Table 2: Re-casting the literature using Rice's framework

Reference	*Domain*	*P*	*A*	*Y*	*F*	*S*
Aha [1992]	classification	75 datasets	3 rule learners	accuracy	7 measures of Statistical distributions	C4.5
Brazdil & Henery [1994]	classification	22 datasets = StatLog	23 machine learning, neural and statistical learning algorithms = StatLog	accuracy	16 measures of statistical distributions and information theory = StatLog	C4.5
Gama & Brazdil [1995]	classification	StatLog	StatLog	accuracy	StatLog	regression
Linder & Studer [1999]	classification	StatLog	StatLog	accuracy	StatLog	CBR
Kalousis & Theoharis [1999]	classification	StatLog	StatLog	accuracy	StatLog	NOEMON (pairwise NNs)
Brazdil et al. [2003]	classification	53 datasets	10 machine learning algorithms	accuracy and time	StatLog	k-nearest neighbor ranking
Lim et al. [2000]	classification	32 datasets	33 machine learning algorithms	accuracy, time, tree size	StatLog	C4.5
Smith et al. [2001]	classification	57 datasets	6 machine learning algorithms	accuracy	21 measures: StatLog plus additional statistical measures	Neural network
Smith et al. [2002]	classification	57 datasets	6 machine learning algorithms	accuracy	21 measures: StatLog plus additional statistical measures	Self-organising feature map
Ali & Smith [2006]	classification	112 datasets	8 machine learning algorithms	accuracy and time	31 measures: StatLog plus additional statistical measures	C4.5
Kopf et al. [2000]	regression	5450 datasets	3 regression models	error	StatLog	C5.0
Ali & Smith-Miles [2006]	kernel selection	112 datasets	5 kernels within SVM	accuracy	29 measures: StatLog plus additional statistical measures	C4.5
Soares et al. [2004]	SVM parameter selection	42 datasets	11 parameters	error	14 statistical features of regression problems	k-nearest neighbor ranking
Arinze et al. [1997]	forecasting	67 time series	6 forecasting methods	error	6 features	Expert system
Venkat. & Sohl [1999]	forecasting	180 series	9 forecasting methods	error	6 features	Neural network
Prudencio & Ludermir [2004]	forecasting	99 series	2 forecasting methods	error	14 features	C4.5
Prudencio & Ludermir [2004]	forecasting	645 series	3 forecasting methods	error	5 features	NOEMON
Wang [2005]	forecasting	315 series	4 forecasting methods	error	13 features	C4.5 and SOFM
Lagoudakis et al. [2001]	sorting	10000 sequences	3 sorting algorithms	time	1 feature (length of sequence)	Dynamic programming
Guo [2003]	sorting	43195 sequences	5 sorting algorithms	time	3 measures of pre-sortedness	C4.5 & Bayesian classifier
Horvitz et al. [2001]	constraint satisfaction	5000 QWH problems	2 solvers	time	Statistical features of the problem and solver statistics	Bayesian network
Leyton-Brown et al. [2002]	constraint satisfaction	8544 WDP problems	1 solver (CPLEX)	time	25 Statistical features of the problem and solver statistics	Regression
Leyton-Brown et al. [2003]	constraint satisfaction	4500 WDP problems	3 solvers	time	25 Statistical features of the problem and solver statistics	Regression
Nudelman et al. [2004]	constraint satisfaction	20000 SAT problems	3 solvers	time	91 Statistical features of the problem and solver statistics	Regression
Xu [2007]	constraint satisfaction	4811 SAT problems	7 solvers with & w/o preprocessing	time	48 features of the problem and solver statistics	Regression
Samulowitz [2007]	constraint satisfaction	1697 QBF problems	10 branching heuristics	time	78 features of the problem and solver statistics	Regression
Smith-Miles [2007)	optimization	28 QAP problems	3 meta-heuristics	objective function gap	8 measures of problem size and complexity, and search space characteristics	Neural network and SOFM

5. EXTENSIONS

Beyond the Algorithm Selection Problem, there are many ideas under the banner of meta-learning that can be adopted by researchers from other disciplines. While it is beyond the scope of the current paper to review these methods, the interested reader may follow up on some of the concepts that could find generalisation beyond machine learning, including: combining algorithms and voting schemes (boosting, bagging, stacked generalisation, mixture of experts modelling – see for example Wolpert [1992]; Chan and Stolfo [1997]; Opitz and Maclin [1999]; Gama and Brazdil [2000]; Todorovski and Dzeroski [2003]; Peterson and Martinez [2005]), algorithm portfolios [Gagliolo and Schmidhuber 2006], landmarking [Pfahringer et al. 2000; Ler et al. 2005], dynamic algorithm selection [Armstrong et al. 2006; Samulowitz and Memisevic 2007] and real-time analysis of algorithms, particularly dynamic tuning of parameters via racing algorithms [Maron and Moore 1997].

Similarly, there are many ideas developed in the AI community that could be worth generalising to other disciplines. Wallace and Schimpf [2002] have explored automatic algorithm design, seeking to construct an optimal algorithm for different classes of constrained optimisation problems by searching through the space of hybrid algorithms. The search space includes choice of problem variables and constraints (problem formulation) as well as the search algorithm and defining behaviours. The theme of algorithm design, rather than selection, has also been approached by Leyton-Brown et al. [2003b], who adopt the concept of boosting from the machine learning community and apply similar principles to the design of algorithm portfolios. The key is to identify the uncorrelated regions where each algorithm is most effective and construct portfolios that span this space, focusing on new algorithm design in the regions that the portfolio finds most difficult.

There is also the opportunity to apply Rice's framework for the algorithm selection problem to a wider range of problem domains than have currently been considered. The applicability of the approach, and the resulting (meta-)learning ideas that can be applied extends well beyond the classification, regression, forecasting, sorting, constraint satisfaction, and optimization domains discussed in this paper. Learning the algorithm selection problem can be applied in any domain where there is availability of i) many problem instances of diverse complexities (P); ii) a wide choice of existing algorithms with diverse behaviours (A); iii) clear metrics of algorithm performance (Y); and iv) known features of instances that can be calculated and that correlate with hardness/complexity (F). Undoubtedly the last criteria, the set of features that correlate with complexity, is the most difficult to determine, and the major impediment to straightforward application of meta-learning ideas to other domains. Consider though the domain of financial trading. A number of heuristic rules or algorithms are typically used to determine when to buy and sell stocks, and the characteristics of the trade can be measured by features that capture the order requirements, stock characteristics, and market conditions (see Yang and Jiu [2006]). The relationship between these characteristics and the performance of different algorithms can then be learned. Meta-learning concepts have recently been extended to the selection of response automation strategies in a help-desk domain [Marom et al. 2007]. Other problem domains that seem like natural contenders for the adoption of such an approach include selection of data compression algorithms; bioinformatics algorithms for sequence alignment, gene prediction, protein identification, pattern matching, etc. (see for example, Jones and Pevzner [2004]); cryptography; clustering; matrix inversion algorithms; etc. For many of these domains measuring the characteristics of the problem instances will rely heavily on information theory, as we have seen for classification problems too, but there will also be more domain-specific metrics that need to be developed to ensure strong characterisation of problem instance complexity.

As a further extension of the ideas discussed in this paper, beyond developing automated algorithm selection tools, there is also a need to develop greater insight into the characteristics of problems that lend themselves to easy solution by certain algorithms, and close the loop to feedback this insight to improve algorithm design. The NFL theorem suggests that it is not possible to be able to design a single super-algorithm that performs best for all problem instances, but with greater insight into the conditions under which certain algorithms perform well, existing algorithms can knowledgeably be adapted or new algorithms designed to extend the classes of problems and instances on which we can expect algorithms to perform well.

Regardless of the problem domain, the framework shown in Figure 2 proposes an empirical learning approach to assist with both automated algorithm selection, as well as improved algorithm design. Phase 1 requires the generation of meta-data for the problem domain (the set of P, A, Y, F). Phase 2 focuses on learning of this meta-data to learn the relationship between the instance features (F) and the algorithm performance measures (Y). This will generate empirical rules or algorithm rankings, which then require integration into an automated algorithm selection model. Phase 3 focuses on examining the empirical results of meta-learning from a theoretical perspective with a view to confirming the sense or otherwise of the rules, as well as generating insights into algorithm behaviour that can be used to refine the algorithms. This essentially closes the loop when the performance of the refined (perhaps hybrid) algorithms is evaluated.

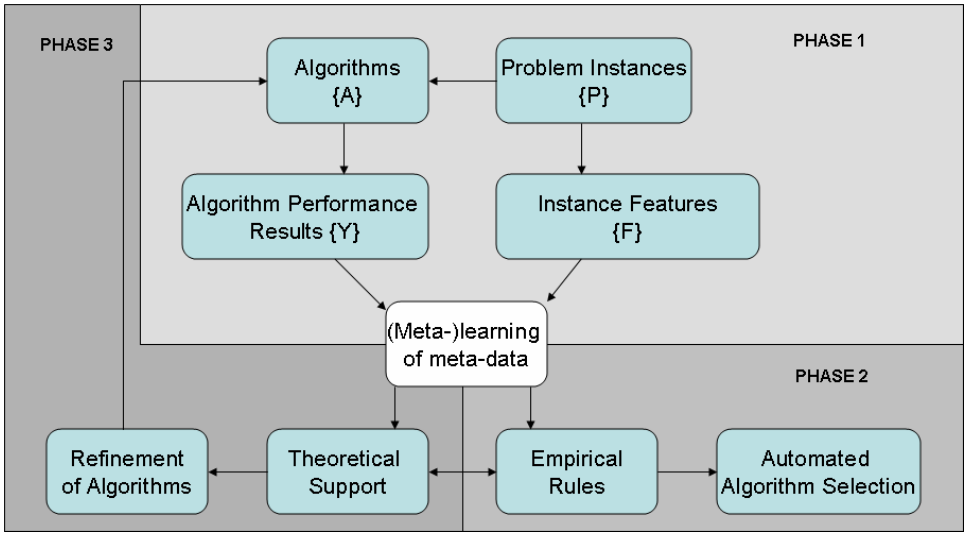

Figure 2: Proposed framework for achieving automated algorithm selection and improved algorithm design via learning of meta-data

6. CONCLUSIONS

This paper has discussed the Algorithm Selection Problem, originally formulated by Rice in 1976, and has demonstrated that many different directions have been followed by different communities seeking guidance on how to select the best algorithm for their problem domain. The task of automated algorithm selection lies at the intersection of many disciplines: computational complexity theory, algorithmic information theory, artificial intelligence, machine learning, operations research, to name a few. It is surprising how little intersection there has been in the relevant developments in these different communities, and how the vocabulary has evolved in different directions, making searching for relevant papers in other disciplines more difficult. While the machine learning community developed meta-learning ideas for solving the problem, the vocabulary slowly changed to

describe the task as model selection rather than algorithm selection, with little reference to Rice [1976] in the process. The AI community developed methods of algorithm selection without much acknowledgement of the meta-learning ideas and their possible generalisation. The metaheuristics community within the Operations Research discipline has developed tools such as landscape analysis and studied search space characteristics without linking to the idea of learning the relationships between these characteristics and algorithm performance, and without much acknowledgement of the AI community's efforts in related ideas for constraint programming.

There is no doubt that all of these communities would benefit from a greater awareness of the achievements in various cross-disciplinary approaches to algorithm selection. This would open up opportunities for extension to both new problem domains as well as new methodologies through cross-fertilization of these ideas. It is intended that this paper will serve as a useful starting point for integrating the different bodies of literature. Using the notation of Rice [1976] the paper has presented a wide cross-section of existing relevant studies from different disciplines using a common framework, exposing the similarities and differences of the approaches, and the opportunities for extension. It is a problem well worth refocusing attention upon, since with the proliferation of new algorithms available each year, Rice's Algorithm Selection Problem is even more relevant today than it was in 1976.

ACKNOWLEDGEMENTS

The author is grateful to the three anonymous referees, whose comments and suggestions helped to improve the paper.

REFERENCES

ACHLIOPTAS, D., NAOR, A. AND PERES, Y. 2005. Rigorous location of phase transitions in hard optimization problems. *Nature.* 435, June, 759-764.

AHA, D. 1992. Generalizing from case studies: a case study. In Proc. 9th Int. Conf. on *Machine Learning*. 1-10.

ALI, S. AND SMITH, K. A. 2005. Kernel width selection for SVM classification: a meta learning approach. *International Journal of Data Warehousing and Mining*. 1, 78-97.

ALI, S. AND SMITH, K. 2006. On learning algorithm selection for classification. *Applied Soft Computing*. 6, 2, 119-138.

ALI, S., AND SMITH-MILES, K. 2006. A meta-learning approach to automatic kernel selection for support vector machines. *Neurocomputing*. 70, 1-3, 173-186.

ANGEL, E. AND ZISSIMOPOULOS, V. 2002. On the hardness of the quadratic assignment problem with meta-heuristics. *Journal of Heuristics*. 8, 399-414.

ARINZE, B. 1995. Selecting appropriate forecasting models using rule induction. *Omega International Journal of Management Science*. 22, 6, 647-658.

ARINZE, B., KIM, S. L. AND ANANDARAJAN, M. 1997. Combining and selecting forecasting models using rule based induction. *Computers and Operations Research*. 24, 5, 423-433.

ARMSTRONG, W., CHRISTEN, P., MCCREATH, E. AND RENDELL, A.P. 2006. Dynamic Algorithm Selection Using Reinforcement Learning, *Proceedings of the International Workshop on Integrating AI and Data Mining* 18-24.

ASUNCION, A AND NEWMAN, D.J. 2007. *UCI Machine Learning Repository* (http://www.ics.uci.edu/~mlearn/MLRepository.html). Irvine, CA: University of California, Department of Information and Computer Science.

BACHELET, V. 1999. *Métaheuristiques Paralleles Hybrides: Application au Probleme D'Affectation Quadratique*. PhD dissertation, Universite des Sciences et Technologies de Lille, France, December.

BENSUSAN, H., GIRAUD-CARRIER, C. AND KENNEDY, C. 2000. A Higher-order Approach to Meta-learning. In *Proceedings of the European Conference on Machine Learning*, Workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination, 109-117.

BENSUSAN, H. AND KALOUSIS, A. 2001. Estimating the predictive accuracy of a classifier. *Lecture Notes in Computer Science*, 2167, 25-31.

BERNSTEIN, A., PROVOST, F., AND HILL, S. 2005. Toward intelligent assistance for a data mining process: an ontology-based approach for cost-sensitive classification. *IEEE Transactions on Knowledge and Data Engineering*. 17, 4, 503-518.

BERRER, H., PATERSON, I. AND KELLER, J. 2000. Evaluation of machine-learning algorithm ranking advisors. In Pavel Brazdil and Alipio Jorge (eds.), *Proceedings of the PKDD-00 Workshop on Data Mining, Decision Support,Meta-Learning and ILP: Forum for Practical Problem Presentation and Prospective Solutions*, Lyon, France.

BORENSTEIN, Y. AND POLI, R. 2006. Kolmogorov complexity, optimization and hardness. IEEE Congress on Evol. Comput.. 112-119.

BOUKEAS, G., HALATSIS, C., ZISSIMPOLOULOS, V. AND STAMATOPOULOS, P. 2004. Measures of instrinsic hardness for constraint satisfaction problem instances. *Lecture Notes in Computer Science*, 2932, 184-195.

BRAZDIL, P. AND HENERY, R. 1994. Analysis of Results. in Michie, D., Spiegelhalter, D.J. and Taylor, C.C. (eds). *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, New York. Chapter 10.
BRAZDIL, P., SOARES, C. AND COSTA, J. 2003. Ranking learning algorithms: using IBL and meta-learning on accuracy and time results. *Machine Learning*. 50, 3, 251-277.
BRODLEY, C. E. 1993. Addressing the selective superiority problem: automatic algorithm/model class selection. *Proceedings of the 10th International Machine Learning Conference*. Amherst, MA. 17-24.
BURKE, E., KENDALL, G., NEWALL, J., HART, E., ROSS, P. AND SCHULENBURG, S. 2003. Hyper-heuristics: an emerging direction in modern search technology. In Glover and Kochenberger (eds.), *Handbook of Metaheuristics*, Kluwer Academic Publishers, Dordrecht, 457-474.
CASTIELLO, C., CASTELLANO, G. AND FANELLI, A. M. 2005. Meta-data: characterization of input features for meta-learning. *Lecture Notes in Artificial Intelligence*, 3558, 457-468.
CHAN, P. AND STOLFO, S. J. 1997. On the accuracy of meta-learning for scalable data mining. *Journal of Intelligent Information Systems*. 8, 1 9, 5-28.
CHU, C. H. AND WIDJAJA, D. 1994. A neural network system for forecasting method selection. *Decision Support Systems*. 12, 13-24.
DESJARDINS, M AND GORDON, D. F. 1995. Special issue on "Bias evaluation and selection". *Machine Learning*. 20, 1-2.
DUCH, W. AND GRUDZINSKI, K. 2001. Meta-learning: searching in the model space. In Proc. *Int. Conf. on Neural Information Processing* (ICONIP). I, 235-240.
ESTIVILL-CASTRO, V. AND WOOD, D. 1992. A survey of adaptive sorting algorithms. *ACM Computing Surveys*. 24, 4, 441-476.
FREY, P.W. AND SLATE, D.J. 1991. Letter recognition using holland-style adaptive classifiers. *Machine Learning*, 6, 161–182.
FURNKRANZ, J. AND PETRAK, J. 2001. An evaluation of landmarking variants. In C. Giraud-Carrier, N. Lavrac, Steve Moyle, and B. Kavsek, (eds.), *Proceedings of the ECML/PKDD Workshop on Integrating Aspects of Data Mining, Decision Support and Meta-Learning (IDDM-2001)*, 57-68, Freiburg, Germany.
GAGLIOLO, M. AND SCHMIDHUBER, J. 2006. Learning dynamic algorithm portfolios. *Annals of Mathematics and Artificial Intelligence*. 47, 3-4, August, 295-328.
GAMA, J. AND BRAZDIL, P. 2000. Cascade generalization. *Machine Learning*. 41, 3, 315-343.
GAMA, J. AND BRAZDIL, P. 1995. Characterization of classification algorithms. In *Proc. 7th Portugese Conference in AI*. 83-102.
GLOVER, F. AND KOCHENBERGER, G. 2003. *Handbook of Metaheuristics*. Kluwer Academic Publishers, Dordrecht.
GNANADESIKAN, R. 1997. *Methods for Statistical Data Analysis of Multivariate Observations*, 2nd Ed. Wiley, New York.
GUO, H. 2003. *Algorithm Selection for Sorting and Probabilistic Inference: A Machine Learning-Based Approach*. Ph.D. Dissertation, Kansas State University, USA.
HILARIO, M. 2002. Model complexity and algorithm selection in classification. In S. Lange, K. Satoh, and C. H. Smith, (eds.), *Proceedings of the 5th International Conference on Discovery Science (DS-02)*, Lübeck, Germany, Springer-Verlag. 113-126.
HOOS, H. H. AND STÜTZLE, T. 1999. Towards a characterisation of the behaviour of stochastic local search algorithms for SAT. *Artificial Intelligence*, 112, 1-2, 213-232.
HORVITZ, E., RUAN, Y., GOMES, C., KAUTZ, H., SELMAN, B., AND CHICKERING, M. 2001. A Bayesian approach to tackling hard computational problems. In *Proc. 17th Conference on Uncertainty in Artificial Intelligence*.
HYNDMAN, R. *Time Series Data Library*, (2002) available from http://www-personal.buseco.monash.edu.au/~hyndman/TSDL/.
JONES, N. AND PEVZNER, P. 2004. *An introduction to bioinformatics algorithms*, MIT Press, Cambridge, MA.
JONES, T. AND FORREST, S. 1995. Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms. *Proceedings of the International Conference on Genetic Algorithms*, 184-192.
KALOUSIS, A. AND HILARIO, M. 2001. Model selection via meta-learning. *International. Journal on AI Tools*. 10, 4.
KALOUSIS, A. AND THEOHARIS, T. 1999. Design, implementation and performance results of an intelligent assistant for classifier selection. *Intelligent Data Analysis*. 3, 5, 319-337.
KEOGH, E. AND FOLIAS, T. 2002. *The UCR time series data mining archive*. available from http://www.cs.ucr.edu/~eamonn/TSDMA/
KNOWLES, J. D. AND CORNE, D. W. 2002. Towards landscape analysis to inform the design of a hybrid local search for the multiobjective quadratic assignment problem. In Abraham, A., Ruiz-del-Solar, J. and Koppen, M. (eds.), *Soft Computing Systems: Design, Management and Applications.* IOS Press, 271-279, Amsterdam.
KOHONEN, T. 1988. *Self-Organisation and Associative Memory*. Springer-Verlag, New York.
KOPF, C., TAYLOR, C. AND KELLER, J. 2000. Meta-analysis: from data characterisation for meta-learning to meta-regression. In Brazdil, P. and Jorge, A. (ed.) *PKDD'2000 Workshop on Data Mining, Decision Support, Meta-Learning and ILP*.
KUBA, P., BRÁZDIL, P., SOARES, C. AND WOZNICA, A. 2002. Exploiting sampling and meta-learning for parameter setting for support vector machines. In Herrera, F., Riquelme, J. C. and Aguilar, J. (Ed.), In *Proc. Workshop Learning and Data Mining associated with Iberamia 2002, VIII Iberoamerican Conference on Artificial Intelligence*. Univ. of Sevilla, Spain, 209–216.
LAGOUDAKIS, M., LITTMAN, M. AND PARR, R. 2001. Selecting the right algorithm. In Prof. *2001 AAAI Fall Symposium Series: Using Uncertainty within Computation*, Cape Cod, MA, November.
LER, D., KOPRINSKA, I. AND CHAWLA, S. 2005. Utilising Regression-Based Landmarkers within a Meta-Learning Framework for Algorithm Selection, *Proceedings of the Workshop on Meta-Learning, 22nd International Conference on Machine Learning (ICML)*, Bonn, Germany, 44-51.
LEYTON-BROWN, K., NUDELMAN, E. AND SHOHAM, Y. 2002. Learning the empirical hardness of optimization problems: the case of combinatorial auctions. *Lecture Notes in Comp. Sci*. 2470, 556-569.
LEYTON-BROWN, K., NUDELMAN, E., ANDREW, G., MCFADDEN, J. AND SHOHAM, Y. 2003. A portfolio approach to algorithm selection. In Proc. *International Joint Conference on Artificial Intelligence*. 2003a, 1542-1543.
LEYTON-BROWN, K., NUDELMAN, E., ANDREW, G., MCFADDEN, J. AND SHOHAM, Y. 2003. Boosting as a metaphor for algorithm design. In Proc. *Principles and Practice of Constraint Programming*. 2003b, 899-903.
LIM, T.-S., LOH, W.-Y. AND SHIH, Y.-S. 2000. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms, *Machine Learning*, 40, 203-228.
LINDER, C. AND STUDER, R. 1999. AST: Support for algorithm selection with a CBR approach. In Proc. *16th Int. Conf. Machine Learning*.
MAKRIDAKIS, S. AND HIBON, M. 2000. The M3-Competition: results, conclusions and implications. *International Journal of Forecasting*. 16, 4, 451-476.
MAROM, Y. ZUKERMAN, I. AND JAPKOWICZ, N. 2007. A Meta-learning approach for selecting between response automation strategies in a help-desk domain. In Proceedings *22nd AAAI Conference on Artificial Intelligence*, 907-912.
MARON, O. AND MOORE, A. W. 1997. The racing algorithm: model selection for lazy learners. *Artificial Intelligence Review*, 193-225.
MERZ, P. 2004. Advanced fitness landscape analysis and the performance of memetic algorithms. *Evol. Comput*. 12, 303-325.

MICHIE, D., SPIEGELHALTER, D.J. AND TAYLOR C.C. (eds) 1994. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, New York.
NUDELMAN, E., LEYTON-BROWN, K., HOOS, H., DEVKAR, A. AND SHOHAM, Y. 2004. Understanding random SAT: beyond the clauses-to-variables ratio. *Lecture Notes in Computer Science*, 3258, 438-452.
OPITZ, D. AND MACLIN, R. 1999. Popular ensemble methods: an empirical study. *Journal of Artificial Intelligence Research*. 11, 169-198.
PENG, Y., FLACH, P., SOARES, C. AND BRAZDIL P. 2002. Improved dataset characterization for meta-learning. In Proc. *Discovery Science, 5th International Conference*. Lübeck, Germany.
PETERSON, A. H. AND MARTINEZ, T. R. 2005. Estimating the potential for combining learning models. In Proc. *ICML Workshop on Meta-Learning.* 68–75.
PFAHRINGER, B., BENSUSAN, H. AND GIRAUD-CARRIER, C. 2000. Meta-learning by landmarking various learning algorithms. In Proc. *17th International Conference on Machine Learning*.
PRODROMIDIS, A. L., CHAN, P. AND STOLFO, S. J. 2000. Meta-learning in distributed data mining systems: issues and approaches. In KARGUPTA, H. and CHAN, P. (Ed.) *Advances of Distributed Data Mining*. AAAI press.
PRUDÊNCIO, R. AND LUDERMIR, T. 2004. Meta-learning approaches to selecting time-series models. *Neurocomputing*. 61, 121-137.
RENDELL, L. AND CHO, H. 1990. Empirical learning as a function of concept character. *Machine Learning*. 5, 267-298.
RICE, J. R. 1953. Classes of recursively enumerable sets and their decision problems. *Transactions of the American Mathematical Society*. 89, 29-59.
RICE, J. R. 1976. The algorithm selection problem. *Advances in Computers*. 15, 65-118.
SAMULOWITZ, H. AND MEMISEVIC, R. 2007. Learning to solve QBF. In Proceedings *22nd AAAI Conference on Artificial Intelligence*, 255-260.
SEEWALD, A., PETRAK, J. AND WIDMER, G. 2001. Hybrid decision tree learners with alternative leaf classifiers: an empirical study. In Proc. *14th International FLAIRS Conference*.
SELMAN, B., MITCHELL, D. G., AND LEVESQUE, H. J. 1996. Generating hard satisfiability problems, *Artificial Intelligence*, 81, 17-29.
SHANNON, C.E. 1948. A mathematical theory of communication. *Bell System Technical Journal*. 27, July & October, 379–423 & 623–656.
SLANEY, J. AND WALSH, T. 2001. Backbones in optimization and approximation. In Proc. *International Joint Conference on Artificial Intelligence*.
SMITH, K. A., WOO, F., CIESIELSKI, V. AND IBRAHIM, R. 2001. Modelling the relationship between problem characteristics and data mining algorithm performance using neural networks. In Dagli, C. et al. (Eds.). *Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining, and Complex Systems*. ASME Press, 11, 356-362.
SMITH, K. A., WOO, F., CIESIELSKI, V. AND IBRAHIM, R. 2002. Matching data mining algorithm suitability to data characteristics using a self-organising map. In Abraham, A. and Koppen, M. (eds.), *Hybrid Information Systems*, Physica-Verlag, Heidelberg, 169-180.
SMITH-MILES, K. A. 2007. Learning the performance of heuristics for the QAP based on search space characteristics. *Deakin Technical Report, Faculty of Science and Technology. TR C07/06*, Deakin University, Australia.
SOARES, C., PETRAK, J. AND BRAZDIL, P. 2001. Sampling-based relative landmarks: Systematically test-driving algorithms before choosing, In Proc. *Portuguese AI Conference*.
SOARES, C., BRAZDIL, P. AND KUBA, P. 2004. A meta-learning method to select the kernel width in support vector regression. *Machine Learning*. 54, 3, 195-209.
STREETER, M., GOLOVIN, D. AND SMITH, S. F. 2007. Combining multiple heuristics online. In Proceedings *22nd AAAI Conference on Artificial Intelligence*, 1197-1203.
STÜTZLE, T. AND FERNANDES, S. 2004. New benchmark instances for the QAP and the experimental analysis of algorithms. *Lecture Notes in Computer Science*, 3004, 199-209.
TELESIS, O. AND STAMATOPOULOS, P. 2001. Combinatorial optimization through statistical instance-based learning. Proceedings of the *13th International Conference on Tools with Artificial Intelligence*, 203-209.
TODOROVSKI, L., BLOCKEEL, H. AND DZEROSKI, S. 2002. Ranking with predictive clustering trees. In Proc. *European Conference on Machine Learning*.
TODOROVSKI, L. AND DZEROSKI, S. 2003. Combining classifiers with meta decision trees. *Machine Learning*.
VAN HEMERT, J. I. 2006. Evolving combinatorial problem instances that are difficult to solve. *Evolutionary Computation*. 14, 433-462.
VENKATACHALAM, A. R. AND SOHL, J. E. 1999. An intelligent model selection and forecasting system. *International Journal of Forecasting*. 18, 3, 67-180.
VILALTA, R. AND DRISSI, Y. 2002. A perspective view and survey of meta-learning. *Artificial Intelligence Review*. 18, 77-95.
VOLLMANN, T.E. AND BUFFA, E.S. 1966. The facility layout problem in perspective. *Management Science*, 12, 10, 450-468.
WALLACE, C. S. AND BOULTON D. M. 1968. An information measure for classification. *Computer J*. 11, 2, 185-194.
WALLACE, M. AND SCHIMPF, J. 2002. Finding the right hybrid algorithm – a combinatorial meta-problem. *Annals of Mathematics and Artificial Intelligence*. 34, 259-269.
WANG, X. 2005. *Characteristic based forecasting for time series data*, PhD. Dissertation, Monash University, Australia.
WANG, X., SMITH, K. A., AND HYNDMAN, R. 2006. Characteristic-based clustering for time series data. *Data Mining & Knowledge Discovery*. 13, 335-364.
WITTEN, I. H. AND FRANK, E. 2005. *Data Mining: Practical machine learning tools and techniques*, 2nd Edition. Morgan Kaufmann, San Francisco.
WOLPERT, D. AND MACREADY, W. 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*. 1, 67-82.
WOLPERT, D.H. 1992. Stacked Generalization. Neural Networks. 5, 241-259.
XU, L., HUTTER, F., HOOS, H. AND LEYTON-BROWN, K. 2007. Satzilla-07: The design and analysis of an algorithm portfolio for SAT. In Principles and Practices of Constraint Programming. *Lecture Notes in Computer Science*. 2007a, 712-727.
XU, L., HOOS, H. AND LEYTON-BROWN, K. 2007. Hierarchical hardness models for SAT. In Principles and Practices of Constraint Programming. *Lecture Notes in Computer Science*. 2007b, 696-711.
YANG, J. AND JIU, B. 2006. *Algorithm selection: a quantitative approach. Algorithmic Trading II: Precision, Control, Execution*, available on-line from http://www.itg.com/news_events/papers/AlgoSelection20060425.pdf, accessed 22nd February 2007.