Evolutionary Parameter Meta-Optimization in Decision Tree Learning

Evolutionary parameter optimization, grid search, simple genetic algorithm, ID3

Abstract

The process of identifying the optimal parameters for an optimization algorithm or a machine learning one is usually costly and involves the search of a large, possibly infinite, space of candidate parameter set. Our work attempts to explore this research area further by analyzing the behavior of a simple genetic algorithm when used to find the optimal parameter setting for an ID3 like learner operating on given datasets.

1 Introduction

This study evidences the necessity for academic research about optimization methods and machine learning systems to provide detailed accounts about how the parameters of the systems have been determined because the experimental results may vary widely when different values for the parameters are employed [Blum and Roli, 2003; Goldberg, 1989; Smith-Miles, 2008; Reif *et al.*, 2012; Pederson and Chipperfield, 2009; Grefenstette, 1986; Neri, 2012].

We also point out that good parameter values are problem dependent. Thus methodologies to detemine optimal parameter settings given a machine learning algorithm and a dataset as in [Hutter *et al.*, 2009] deserve more attention from the research community.

In the experimental part of this paper, we show how the performances of a decision tree learner vary widely on a given dataset when its parameters change. We then investigate the capability of a simple Genetic Algorithm (SGA) [Goldberg, 1989], used as a meta-optimizer, in finding good parameters for an ID3 like decision tree learner [Quinlan, 1986].

The long term goals of our reseach are 1. to understand the relationships, if any, between a good set of parameter values and a given machine learning system for a given data set. 2. to explore ways to discover a good enough parameter set, if it exists, by exploiting the relationship of point 1.

For the sake of completeness, we also mention that the research line of the work reported in this study is also known as parameter optimization via meta learning. The objective of the meta-optimization task is to discover the best possible set of parameter values for a given machine learning algorithm when applied to a given learning problem (dataset).

Finally, note that our research does not aim to invalidate previous experimental work. We are well aware that researchers who have been going through the process of manually discovering a good enough set of values for their parameters may not realise that they themselves have acted as "human optimizers". We believe our work merit is in directing some more light on the important facet of parameter selection for the learning algorithm which is an integral part of solving learning problems.

2 Previous studies on parameter optimization

Previous works on parameter optimization as well as results from those studies confirm that learning performances vary widely if the parameter settings changes even on the same dataset. For instance, in [Eiben *et al.*, 1999; Eiben and Smit, 2011] the authors discuss the effect that parameters have on the performance of the Evolutionary Algorithms like the population size, the selection method, the crossover, and mutation operators.

Researchers have tried to classify research studies in meta optimization of learning parameters using a taxonomy [Eiben *et al.*, 1999] like the one in Figure 1 [Eiben *et al.*, 1999] which distinguishes between parameter selection done before running a machine learning system 'parameter tuning' or while the learning process is occurring 'parameter control'. Unfortunately, even though the taxonimy may suggest a full understanding of the problems and a variety of solutions to deal with it, the reality is that the entries in the taxonony only express ideal methods whose concrete implementation is left to future research.

The parameter selection method by tuning is of primary interest to this study. This approach can be further differentiated into parameter selection by trial and error, by analysis or algorithmically. In this study two algorithmic approaches, a brute force one and a metaheuristic one, are compared.

In one study the authors try to use case based reasoning applied among datasets to preselect good parameter settings for a machine learning system [Reif *et al.*, 2012]. We, on the other hand, believe that each datset requires specific parameter optimization for a given learning systems. Also we believe that the work in [Reif *et al.*, 2012] is impractical as it would require the existence of a database of several optimized $< datasets, parametersettings >$ pairs to allow

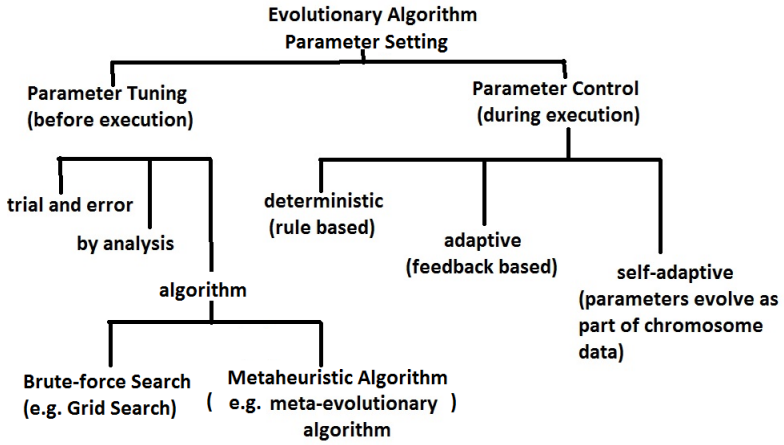

Figure 1: Taxonomy of meta optimization methods

case based reasoning to be applied to select a promising parameter settings for a novel dataset. Our approach extends the taxonomy in an orthogonal way because we make explicit that the dataset under study will influence the performances of the learning algorithms as well as the values of the learning parameters.

The terminology that we use throughout the paper to refer to the main elements of a meta-optimization task is: the given learning problem/dataset is called the *Base learning problem*, the given learning algorithm L1 will be identified by the *Base learning algorithm*. The *meta-optimization problem* consists of finding the best possible parameter setting for L1. The *meta optimisation algorithm* is a machine learning algorithm L2 whose task is to solve the meta-optimisation problem. In Figure 1 , a graphical representation of the meta-optimization task is reported [Neumiller *et al.*, 2011]. In the paper, L1 will be a decision tree learner and L2 will be a simple GA.

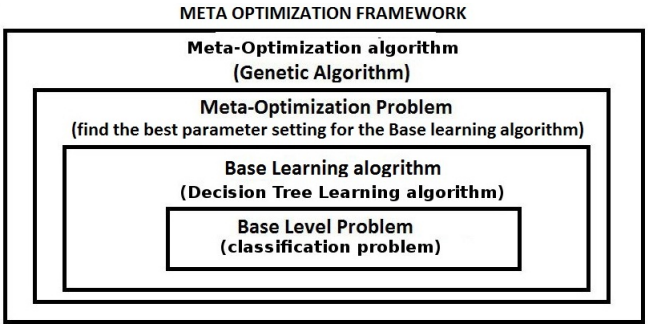

Figure 2: The meta-optimization framework.

3 Our Meta-optimization Methodology

In our approach to the meta-optimization task, a classification problem was selected together with a learning algorithm (a decision tree learner for this study) and we undertook the task to determine the parameter setting for the learning algorithm that will produce models (decision trees in this case) with the lowest error rate or highest accuracy on unseen data.

The decision tree learner that we will use is based on ID3[Quinlan, 1986] whose parameters are: the *Minimum Gain* at which to split a node, the *Maximum Depth* the tree can grow to, the *Minimum Cases* to allow a split.

The algorithm to be used as metaoptimizer is a simple GA. The SGA evolves a population of individuals each of them codifying for a candidate parameter set for the decision tree learner. The fitness value of each chromosome is given by the accuracy value obtained by models generated by the decision tree when run with that specific parameter set.

A statistically valid accuracy value is obtained by averaging the performances obtained from 10 runs of the decision tree learner on different partitions (learning set, testing set) of the dataset maintaining the parameter set constant. In order to explore further the changes in accuracy due to the varying amount of information provided to the learner, each experiment was run with three different partition percentages of the dataset: a) 30% training set and 70% test set; b) 50% training set and 50% test set; and c) 70% training set and 30% test set.

4 Results of the Experiments

The datasets chosen for the experimentation were the*Australian Credit Card Approval* dataset (CCA) and the *Indian Liver Patients* dataset (ILPD).

The *Credit Card Approval* (CCA) dataset contains 690 records of persons. Each instance is described by 15 personal attributes whose meaning was recoded to maintain privacy and a classification attribute indicating whether the applicant had been approved or not. The classification problem is to learn from the available data when to classify an unseen credit card applicant as approved or not.

The *Indian Liver Patients* dataset (ILPD) contains 583 instances representing 416 records of liver patient records and 167 of non liver patient records collected from north east of Andhra Pradesh, India. The classification problem is to learn

from the available 10 attributes when to classify an unseen patient record as a liver patient or not.

The ILPD and CCA datasets were sourced directly from the UCI repository [Frank and Asuncion, 2010]

5 Grid search as a base line parameter meta optimizer

For baseline purposes, we started the experimentation session by running Grid Search (as the meta optimization algorithm) over the parameter space of the decision tree learner to try and assess the overall shape of the accuracy function for any point in the space.

The Grid Search algorithm performed a uniform coverage of the parameter space by sampling the parameter space with a given incremental step that we selected to be small enough to cover as many of the values in each of parameters as was possible given the the amount of computational time and resources that we had available for covering the parameter space.

The size of step was a compromise between covering all the possible values for a parameter and dealing with the combinatorial explosion of parameter sets resulting from exploring every combination of parameter values, particularily continuous valued ones.

For each Grid search the range for the Maximum tree depth was set from 0 to 16 (for the CCA dataset and from 0 to 11 for the ILPD dataset, in steps of 1. The Minimum Information Gain for split was set from 0 to 1 in steps of 0.1 and the Minimum number of Examples for splut was set from 1 to 101 in steps of 10.

These settings resulted in a uniform point cover of 2057 and 1452 different parameter sets for the CCA and ILPD datasets respectively. Three searches were run each time using a different train/test ration and tested the resulting trees on ten randomly selected train/test partitions each time. This resulted in a total of 61710 and 43560 ID3 evaluations for the CCA and ILPD datasets respectively.

6 The accuracy landscape produced by a grid search exploration of the parameter space

In this section we study the accuracy landscape produced by a grid search exploration of the parameter space. The objectives of the experiments were twofold: firstly we wanted to provide a baseline for the meta optimization algorithm, secondly we wanted to illustrate the 'ruggedness' of the accuracy function produced by the space of parameter sets input into the decision tree learner.

In Figure 3 and Figure 4, the accuracy function as estimated by the grid search meta optimizer is reported for the CCA dataset with (70%-30% train/test partition. Each point in the figures reports the ID3 accuracy for each of parameter sets generated by the Grid Search Algorithm. The accuracy value reported by each point is the average accuracy evaluated over the 10 data samples.

It is important to bear in mind that while the grid search algorithm may allow for a uniform coverage of the parameter space, not all the possible combinations of the parameters can

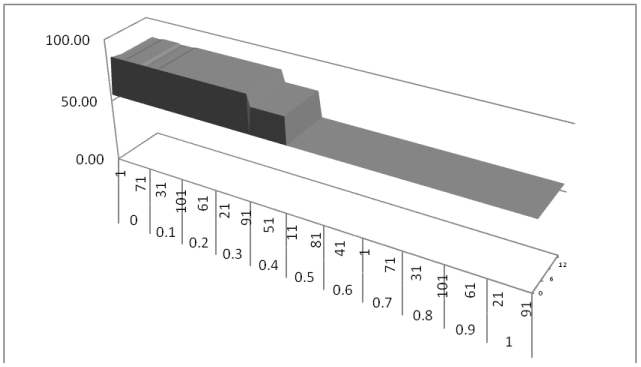

Figure 3: Accuracy function over the parameter space with a 70%-30% (Learning - Test) split of the CCA dataset obtained by using a grid search meta optimizer.The vertical axis report the accuracy value, whilst the left horizontal axis reports two parameter ranges, the Minimum Gain for Split (left outer) and the Minimum Examples for Split (left inner). The right horizontal axis represents the Maximum Tree Depth ID3 was allowed to grow.

be tested, for reasons previously discussed, we have no way to know how the accuracy function behaves for parameter sets in the unevaluated regions.

Sometimes the assumptions of continuity and of linear/planar interpolability among points is made for the accuracy function thus research works report the accuracy function as an irregular landscape like the one that can be seen in Figure 3. We have however to keep in mind that even though the continous landscape style of graphs may be aesthetically appealing and may provide an easy way for the reader to appreciate the overall behavior of the accuracy function. Those latter type of graphs are analytically incorrect. The correct style to be used for reporting the accuracy function is one that accounts for gaps in the region of the parameter space such is done in Figure 4.

The results of the CCA experiments for the 70%/30% train/test partition, as illustrated in Figure 3 and Figure 4 , demonstrate that the highest value of accuracy corresponded to low values (0.0-0.3) of the Minimum Gain for Split parameter. The resulting landscape appears to be a stepped progression from low to high Minimum Gain values. The accuracy across different Minimum Examples does not vary. Increasing Minimum tree depth did not apper to affect accuracy.

The results of the ILPD experiments for the 70%/30% train/test partition, as illustrated in Figure 7, showed that low values (0.0-0.2) of the Minimum Gain for Split parameter similar to the CCA experiments. However, only the lowest values for Minimum Examples for Split (value 1) gave the highest accuracy (unlike CCA). All the other regions of parameter space exhibited continuous planes of lower accuracy.

The resulting landscape shows regions of high accuracy in the low parameter values with regions of lower accuracy in the higher parameter value regions. A further observation is that the accuracy values are highest in the lower values for Minimum Gain. Increasing Minimum tree depth did not appear to affect accuracy.

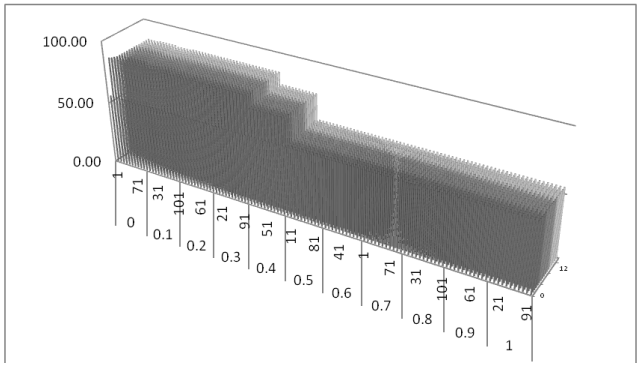

Figure 4: Accuracy function over the parameter space with a 70%-30% (Learning - Test) split of the CCA dataset obtained by using a grid search meta optimizer.The vertical axis report the accuracy value, whilst the left horizontal axis reports two parameter ranges, the Minimum Gain for Split (left outer) and the Minimum Examples for Split (left inner). The right horizontal axis represents the Maximum Tree Depth ID3 was allowed to grow.

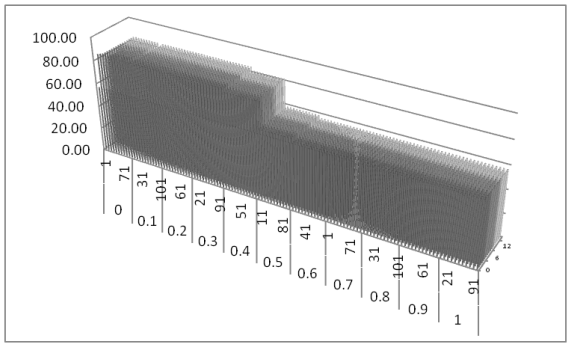

Figure 5: Accuracy function over the parameter space with a 50%-50% (Learning - Test) split of the CCA dataset obtained by using a grid search meta optimizer.

Furthermore Figure 4 and Figure 7 evidence the different Accuracy Landscapes for the two datasets and support the idea that optimal parameter settings for a learning algorithm like ID3 cannot be generalized for different datasets.

Furthermore Figure 5 and Figure 6 showing the CCA Accuracy Landscape on smaller train/test partitions show an overall similarity with the 70%/30% partition experiment. However small differences can be noted, which shows that even the choice of training/test partition size of the same dataset can give different results (see Table 1) and overall behaviour.

7 Experiments using the Simple Genetic Algorithm

We selected the SGA as meta optimizer for this group of experiments as it is known that genetic algorithms are very good as function optimizer [Goldberg, 1989; Michalewicz, 1996]. Thus we want to explore how much a simple heuristic like a

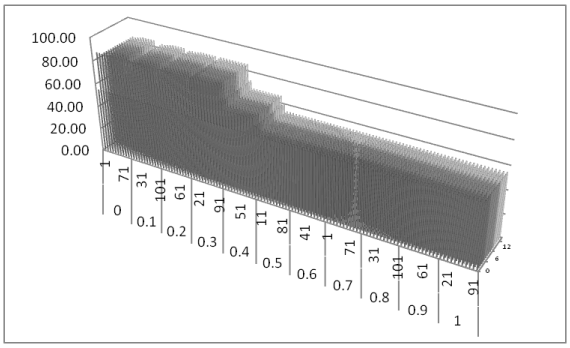

Figure 6: Accuracy function over the parameter space with a 30%-30% (Learning - Test) split of the CCA dataset obtained by using a grid search meta optimizer.

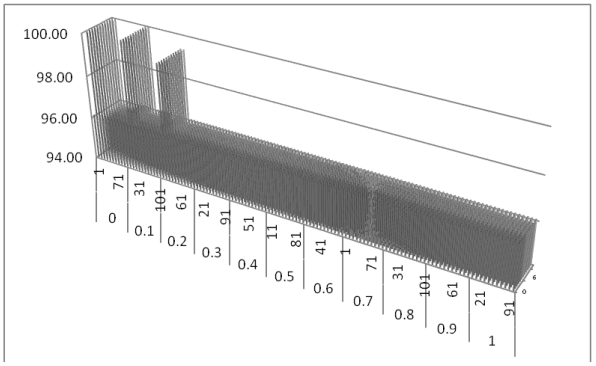

Figure 7: Accuracy function over the parameter space with a 70%-30% (Learning - Test) split of the ILPD dataset obtained by using a grid search meta optimizer.

SGA can improve the search of the parameter space over the grid search heuristic.

The SGA was run in the same experimental setups as those described for the grid search in the previous section. The SGA was run with the following values for its main parameters: population size set at 40, crossover rate set at 0.25, mutation rate set at 0.01, generation number set at 100. Each individual of the population is a binary string that codes for the input parameter of the ID3 algorithm represented with the same ranges and discretizations (steps) used for the grid search in order to make meaningful the comparison of the experiments between grid search and SGA.

Figure 8 and Figure 9 show the typical results of the exploration of the ID3 parameter space using the SGA on both datasets. At a first glance we can observe that the SGA is more effective in exploring the parameter space as not all the points (the missing columns in the graphs) in the parameter space have been explored while still discovering parameter sets in the optimal regions of the parameter space.

Table 1 and Table 2 show the maximum ID3 accuracy evaluated in both Grid Search and SGA experiments on the CCA and ILPD datasets. The SGA discovered the same maximal values for the same train/test splits for both datasets.

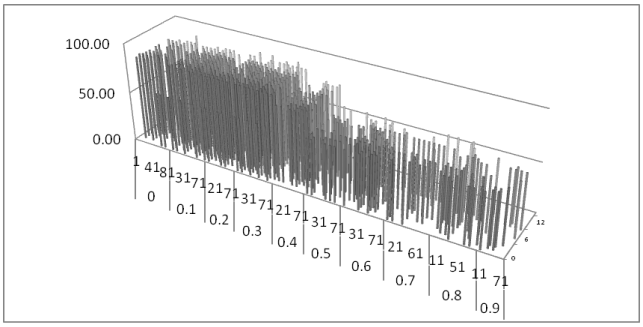

Figure 8: Accuracy function over the parameter space with a 70%-30% (Learning - Test) split of the CCA dataset obtained by using the SGA meta optimizer.

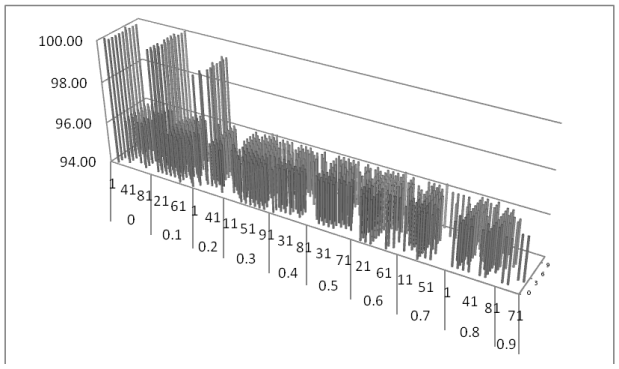

Figure 9: Accuracy function over the parameter space with a 70%-30% (Learning - Test) split of the CCA dataset obtained by using the SGA meta optimizer.

Furthermore, the SGA was run for 5 times using the same parameters on the ILPD dataset. This was done in order to measure the SGA's efficiency in discovering the optimal parameter set. The epoch at which the optimal parameter set was found was averaged across the 5 runs for each training/test partition. Table 2 shows that the average epoch number for the 30%/70%, 50%/50% and 70%/30% training /test partitions was 1, 1.2 and 2.6 respectively. This shows that the SGA was efficient in finding the optimal parameter set in the case of the ILPD dataset, in that the optimal parameter set was discovered at a relatively lower processing cost.

This early discovery of the maximal value may however be more attributable to the size and randomness of the intial populations and the nature of the datasets themselves rather than the evolutionary approach of the meta-optimizer.

8 Comparison with other classifiers

In [Michie *et al.*, 1994] a number of classifiers including CAL5, C4.5, k-NN and Naivebay were applied to the CCA dataset with error rates of 0.131, 0.155, 01.181 and 0.151 respectively. These results are equivalent to an accuracy of 86.9%, 84.5% 81.9% and 84.9% respectively as listed in Table 3. 10-fold cross-validation was used for training and testing. The best results achieved by the SGA experiments for the CCA data set were 85.96%, 84.35% and 85.51% using the 30%/70%, 50%/50% and 70%/30% training /test splits.

Optimizer	Partition	Value
Grid Search	30%/70%	85.96%
	50%/50%	84.35%
	70%/30%	85.51%
SGA	30%/70%	85.96%
	50%/50%	84.35%
	70%/30%	85.51%

Table 1: CCA Maximum Parameter Set Accuracy

Optimizer	Partition	Value	Epoch
Grid Search	30%/70%	100%	
	50%/50%	100%	
	70%/30%	100%	
SGA	30%/70%	100%	1
	50%/50%	100%	1.2
	70%/30%	100%	2.6

Table 2: ILPD Maximum Parameter Set Accuracy

An experiment was run for comparison purposes using the Random Forest and SimpleCart classifiers in the WEKA library [Hall *et al.*, 2009] on the same ILPD dataset using the default parameter settings. An average classification accuracy of 100% was obtained for the same training/test random splits for both classifiers on the ILPD dataset, as shown in Table 4. The best result achieved by the SGA experiments using ID3 for the ILPD data set was also 100/% for the three training /test partitions.

9 Conclusion

In the paper, we have compared Grid Search and SGA as meta optimizers used to find the optimal parameter sets for a ID3 learner used to solve a classification problem.

Grid Search has been used as a base line method to provide coarse but uniform exploration of the parameter space. The SGA heuristic instead has been used to solve in an efficient and effective way the problem to find the optimal parameter sets.

It appears that, on the datasets used in the experimental sessions, the SGA discovered the best parameter sets w.r.t. Grid Search, with a lower computational cost (number of ID3 trees built).

The results suggest that researchers in machine learning or optimization methods that are interested in determining a suitable parameter set for their system could use a SGA heuristic for dealing with the problem in both a formal, structured and efficient way instead of using a manual tuning approach.

SGA optimized ID3)		Other	
train/test split	Accuracy	Classifier	Accuracy
30%/70%	85.96%	CAL5	86.9%
50%/50%	84.35%	C4.5	84.5%
70%/30%	85.51%	k-NN	81.9%
		Naivebay	84.9%

Table 3: CCA-Comparison of ID3 (with SGA optimizer) results with other Classifiers

SGA optimized ID3		Other	
train/test	Accuracy	Classifier	Accuracy
30%/70%	100%	Random Forest	100%
50%/50%	100%	SimpleCart	100%
70%/30%	100%		

Table 4: ILPD Comparison of ID3 (with SGA optimizer) results with other Classifiers

References

[Blum and Roli, 2003] C Blum and A Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35:268–308, 2003.

[Eiben and Smit, 2011] A E Eiben and S K Smit. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation*, page 19 31, 2011.

[Eiben *et al.*, 1999] A E Eiben, R Hinterding, and Z Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3:124 – 141, 1999.

[Frank and Asuncion, 2010] A. Frank and A. Asuncion. Uci machine learning repository, 2010.

[Goldberg, 1989] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Ma, 1989.

[Grefenstette, 1986] J J Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 16:122 128, 1986.

[Hall *et al.*, 2009] M Hall, E Frank, G Holmes, B Pfahringer, P Reutemann, and I H Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11, 2009.

[Hutter *et al.*, 2009] Frank Hutter, Holger H Hoos, Kevin Leyton-Brown, and Thomas Stuutzle. Paramils:an automatic algorithm configuration framework. *Journal of Artificial Intelligence Research (JAIR)*, 6:267–306, 2009.

[Michalewicz, 1996] Z Michalewicz. Genetic algorithms + data structures = evolution programs. *Springer*, 1996.

[Michie *et al.*, 1994] D. Michie, D.J. Spiegelhalter, and C.C. Taylor. Machine learning, neural and statistical classification, 1994.

[Neri, 2012] F. Neri. A comparative study of a financial agent based simulator across learning scenarios. In *ADMI 2011 Agent and Data Mining Interaction*, volume LNAI 7103, pages 86–97. Springer, 2012.

[Neumiller *et al.*, 2011] C Neumiller, S Wagner, G Kronberger, and M Affenzeller. Parameter meta-optimization of metaheuristic optimization algorithms. 2011.

[Pederson and Chipperfield, 2009] M E H Pederson and A J Chipperfield. Simplifying particle swarm optimization. *Applied Soft Computing*, 10:618–628, 2009.

[Quinlan, 1986] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[Reif *et al.*, 2012] M Reif, F Shafait, and A Dengel. Meta-learning for evolutionary parameter optimization of classifiers. *Machine Learning*, 87:357–380, 2012.

[Smith-Miles, 2008] K A Smith-Miles. Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys*, 41:Art 6, 2008.