Parameter Optimization in Decision Tree Learning by using Simple Genetic Algorithms

MICHEL CAMILLERI
University of Malta
Faculty of ICT
Msida
MALTA
michel.camilleri@um.edu.mt

FILIPPO NERI
University of Naples
Dept. of Electrical Engineering
and Information Technology (DIETI)
University of Napoli ”Federico II”
Naples, Italy
filippo.neri@unina.it

Abstract: The process of identifying the optimal parameters for an optimization algorithm or a machine learning one is a costly combinatorial problem because it involves the search of a large, possibly infinite, space of candidate parameter sets. Our work compares grid search with a simple genetic algorithm when used to find the optimal parameter setting for an ID3 like learner operating on given datasets.

Key–Words: Machine Learning, Evolutionary Algorithms, Parameter Optimization

1 Introduction

This study would like to stress the necessity for studies 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 significantly when different values for the parameters are employed [1, 2, 3, 4, 5, 6, 7, 8].

We also believe that it is important to stress the fact that good parameter values are learning system and problem dependent. Thus, methodologies used to determine optimal parameter settings given a machine learning algorithm and a specific dataset, as in [9], deserve more attention from the research community. The No Free Lunch Theorems [10] states that if a certain algorithm performs well on a certain class of problem it pays for that with a degraded performance on the set of the remaining problems. This result is at the core of the experimentation in our study and it is the reason why we are considering metaoptimization as a task has to be done considering both a machine learning system and a dataset.

In the experimental part of this paper, which extends our work in [11, 12, 13] we show how the performances of a decision tree learner vary on a given dataset when its parameters change. We then investigate the capability of a simple Genetic Algorithm (SGA) [2], used as a meta-optimizer, of finding good parameters for an ID3 like decision tree learner [14]. The choice of SGA as the metaoptimizer is based on the fact that it is well documented and understood [15, 16, 17]

The long term goals of our research are to understand the relationships, if any, between a good set of parameter values and a given machine learning system for a given data set and to explore efficient ways to discover a good enough parameter set, if it exists, by exploiting the relationship. This paper presents a graphical method to explore these multidimensional relationships through Accuracy Landscape graphs. These type of charts present the variation of the machine learner’s performance over more than two parameters. The metaoptimization approach to parameter selection of machine learning algorithms using the ’black box’ approach will enable the application of the method to a much wider set of algorithms than other metaoptimizers which are adapted to specific algorithm types. This approach will provide a generally applicable tool which would be easy to use on any machine learner especially novel techniques still in development and whose behaviour and general performance is still to be discovered. Such an approach will contribute to provide a tool by which developers of new algorithms can document clearly the parameters selected for the algorithms and the method used to select them. This makes the parameter selection process more transparent and reproducible.

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

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 [18, 19] 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 preparing several abstract classifications of ideal methods [18] whose concrete implementation is left to future research.

Our approach extends past approaches in an orthogonal way because it is a concrete methodology and 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.

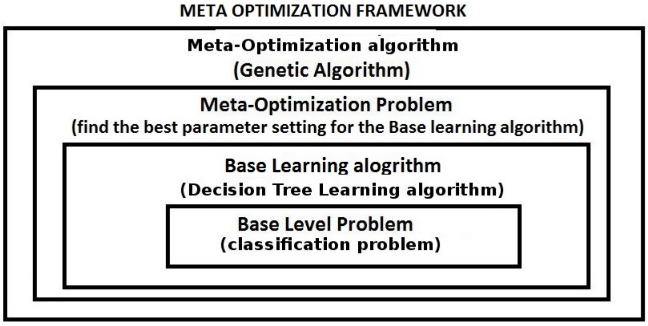
META OPTIMIZATION FRAMEWORK
Meta-Optimization algorithm
(Genetic Algorithm)
Meta-Optimization Problem
(find the best parameter setting for the Base learning algorithm)
Base Learning alogrithm
(Decision Tree Learning algorithm)
Base Level Problem
(classification problem)

Figure 1: The meta-optimization framework.

An example of the current interest in the meta optimization of the learning parameter is a paper just published in the Machine Learning journal where the authors try to use case based reasoning applied among datasets to preselect good parameter settings for a machine learning system [4]. We do not agree however with the underlying philosophy of their work because we believe that each datset requires specific parameter optimization for a given learning systems. Also the work in [4] is impractical as it would require the existence of a database of several optimized $< datasets, parametersettings >$ pairs to allow the case based reasoning approach to be applied to select a promising parameter settings for a novel dataset.

In [9], a metaoptimization approach which starts off with a candidate parameter set and makes single stepwise changes to individual parameters is reported. Our approach, on the other hand, starts off with a population of candidate solutions. Overall improvement in their methodology happens sequentially, whilst in our methodology, optimization happens in parallel.

In [20, 21, 22, 23], a family of optimization approaches which are based on an initially large set of parameter sets. In [20, 22, 23] a Student T-test/Friedman's test is used to eliminate the significantly worse candidates. The method searches the parameter space of an algorithm by doing a broad sweep as possible in the first step. This process of evaluation and elimination is repeated until one candidate, the best, remains. Our methodology based on evolutionary computation uses a set of candidate parameter sets whose size is kept constant through the process.

In [24] the authors compare the effectiveness of optimization using manual, Grid Search and Random Search. The main point of the study was that Random Search can be better than Grid Search on problems with low effective dimensionality. Learning problems with low effective dimensionality have some of their parameters which may not have an affect on the algorithm's performance when varied. We are not introducing any restrictions on our metaoptimization problem and we aim to develop a methodology which is applicable to any Machine Learning system and learning problem.

The terminology that we will use through 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 Base learning algorithm. The meta-optimization problem consists of finding the best possible parameter setting for L1. The meta learning optimisation algorithm is a machine learning algorithm L2 whose task is to solve the meta-optimisation problem. Fig. 1 is a graphical representation of the meta-optimization task [25]. In the paper, L1 will be a decision tree learner and L2 will be a simple GA.

2 Our Meta-optimization Methodology

In our approach to the meta-optimization task, a number of classification problems were selected together with a learning algorithm (a decision tree learner for this study). We then faced the task of determining 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 used was based on ID3 [14]. One of the parameters chosen for optimizaiton was the *Maximum Depth* the tree can grow to, as it was shown to affect ID3 tree performance

[26]. The two other ID3 parameters were the *Minimum Gain* and *Minimum Cases* at which to allow a split to a node. The algorithm used as metaoptimizer was a simple GA [2]. 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 constant the parameter set. The experimental work in this paper complements that of [11].

3 Datasets used

The datasets chosen for the experimentation were the *Adult* dataset (Adult), the *Mushroom* dataset (Mushroom), the *Diabetes* dataset (Diabetes) and the*(Australian) Credit Card Approval* dataset. All the data sets are available directly or indirectly from the UCI repository [27]

The *Adult* dataset (Adult) contains 48842 instances extracted from the census bureau with a mix of discrete and continuous variables. The target variable classifies each instance as to whether the individual earns more than 50K or less than 50K. The classification problem is to learn from the available attributes whether an unseen individual earns more or less than 50K. The original dataset had 14 attributes and the target class. One of the attributes, labelled fnlwgt (final weight), had 21468 different values. This attribute caused the ID3 to generate very large trees which rendered the evaluation process unfeasible in the time available. The field was removed from the data set in order to carry out the study. The accuracy of the resulting experiments still remained relatively high as will be shown later in the paper. So the final number of attributes used in all the experiments on the *Adult* dataset was 13.

The *Mushroom* data set consists of 8124 hypothetical mushroom samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family. Each instance is described by 22 discrete attributes. The target attribute classifies the mushroom instance into definitely edible, definitely poisonous or of unknown edibility. The cases of unknown edibilitgy were re-classified as non-edible. The classification problem was thus reduced to the classification of a previously unseen instance into an edible or non-edible type.

The *Diabetes* dataset contains 768 medical records for patients some of them affected by diabetes. The classification problem is to learn from the available 8 attributes when to classify an unseen patient as suffering of diabetes or not. The dataset was sourced from the WEKA repository at http://www.cs.waikato.ac.nz/ml/weka/ [28]. However it had been originally sourced from the UCI repository [27].

4 The Experiments

The experimental study was carried out in two parts:

The first part was an exploration of how the accuracy of the decision tree learner (ID3 for short) changes when different parameter values are used on each of the three datasets. A Grid Search algorithm was used to generate a broad range of value combinations for three selected ID3 parameters. Each set of parameter values thus generated were input to the ID3 algorithm to generate a classification tree based on a set of training data. Each tree was then tested on test set and the accuracy obtained was recorded together with the parameter values that generated it. The accuracy values were plotted against the parameter set values in the form of a 3-D charts in order to gain insight into the relationship between them.

A broad range of ID3 parameter set values were applied to the Grid Search algorithm to explore as large an area of the parameter space as possible whilst keeping processing cost down by using relatively large step sizes. In the same part of the study the same parameter range/step values were applied to the SGA to examine the ability of the SGA to explore the same ID3 parameter set space at a lower processing cost.

The second part of the study was an attempt to see how the process of searching for the optimimal ID3 parameter set using SGA can vary by modifying one of the SGA's parameters, the Crossover rate that may be one of the main parameters affecting the exploration power of the SGA and performance [17]. In this part of the study the Mushroom and the Diabetes datasets were explored using the SGA with Crossover rates of 25%, 35%, 50% and 70%.

We also point out that all the accuracy values obtained in the following experiments have been obtained by running the decision tree learner (ID3) on 10 different random partitions of the dataset and averaging the accuracy as measured on the test set. Every data point in the following graphs respects this property for both meta optimization algorithms.

In order to explore further the changes in accuracy due to the varying amount of information provided to the learner, each experiment described below was carried out with three training/test 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.

5 Part 1: Initial exploration of Accuracy landscape

For baseline purposes, each experimentation session was started 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 most of the values in each attributes. The size of step results from a compromise between covering all the possible values for an attribute and dealing with the combinatorial explosion of parameter sets resulting from exploring every combination of parameter values. Therefore the step size is determined by the amount of computational time and resources that we have available for covering the parameter space.

The Grid Search algorithm was set with the following range and step sizes for each of the selected ID3 parameters:

- *Maximum tree depth*: for the Adult data set from 0 to 13 (14 values), for the Mushroom dataset from 0 to 22 (23 values), both with step size 1, for the Diabetes data set from 0 to 9 (9 values).

- *Minimum information gain for split*: 11 different values, from 0 to 1, step of 0.1 for all three datasets

- *Minimum number of examples for split*: 11 different values, from 1 to 101, in steps of 10 for all three datasets.

These settings resulted in a uniform point cover of 1694 different parameter sets for the Adult dataset experiments, 2783 for the Mushroom dataset and 1210 for the Diabetes dataset. The evaluation of each parameter set, consisted of the generation and testing of trees by the ID3 algorithm on each of the 10 randomly selected training/test data sets resulting in a total 50820, 83490 and 36300 individual ID3 train and test operations for the three datasets respectively.

As already said the objectives of the reported experiments are twofold: first we want to provide a baseline for the meta optimization algorithm, second we want to convey the view of how rugged is the accuracy function produced by the parameter space when input into the decision tree learner.

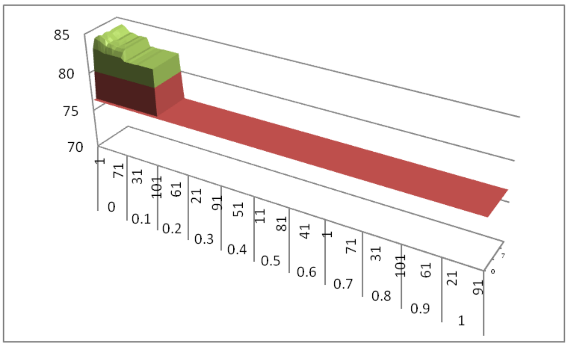

Figure 2: D3 Accuracy function on Adult over the parameter space with a 70%-30% (Learning - Test) split of the datase obtained by using a grid search meta optimizer. The vertical axis report the accuracy value, whilst the left horizontal axis reports two parameter ranges, the Minimum Gain for Split (left outer) and the Minimum Examples for Split (left inner). The right horizontal axis represents the Maximum Tree Depth ID3 was allowed to grow.

6 Results of experiments with SGA

7 Results of Part 1 experiments with Grid Search

Fig. 3, fig. 4 and fig. 5 show the accuracy obtained by running ID3 is reported for each point of the parameter space that have been evaluated. 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 be tested, for reasons previously discussed, thus we have no way to know how the accuracy function behaves in for parameter sets in the unevaluated region.

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 a rugged landscape like the one that can be seen in fig. 2.

We have however to keep in mind that even though the continuous 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 fig. 3.

The Adult experiments showed that low values (0.0-0.2) of the Minimum Gain for Split parame-

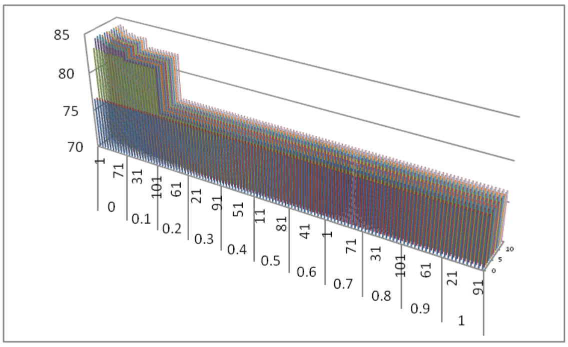

Figure 3: ID3 Accuracy function on Adult over the parameter space with a 70%-30% (Learning - Test) split of the dataset obtained by using a grid search meta optimizer. The vertical axis report the accuracy value, whilst the left horizontal axis reports two parameter ranges, the Minimum Gain for Split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

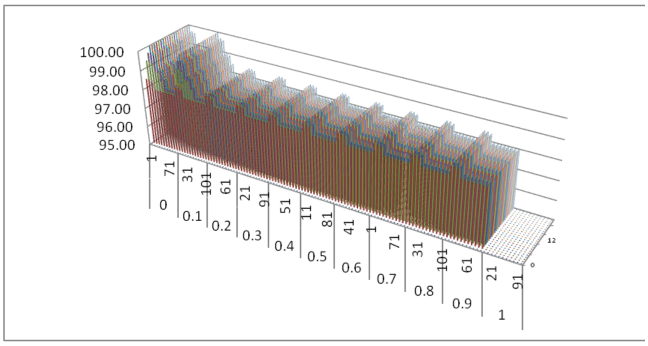

Figure 4: ID3 Accuracy function on Mushroom over the parameter space with a 70%-30% (Learning - Test) split of the datase obtained by using a grid search meta optimizer. The axis report the same parameters as before.

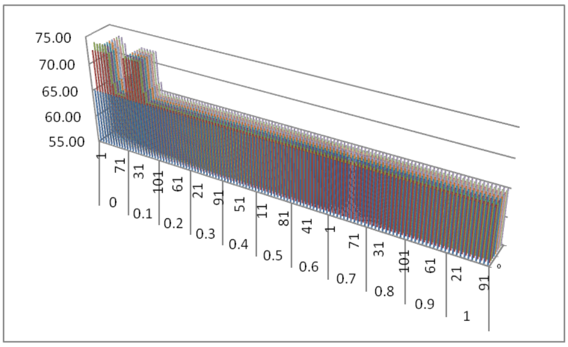

Figure 5: ID3 Accuracy function on Diabetes over the parameter space with a 70%-30% (Learning - Test) split of the datase obtained by using a grid search meta optimizer. The axis report the same parameters as before.

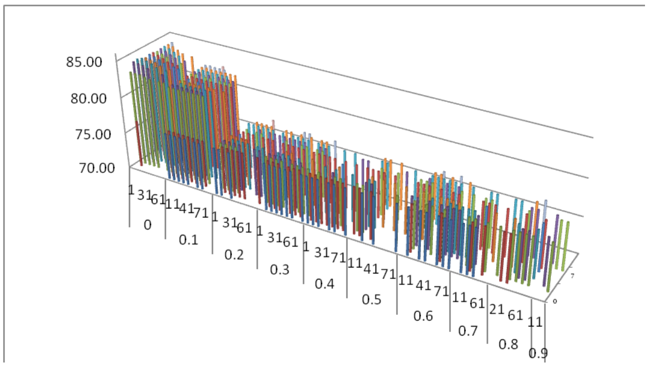

Figure 6: Exploration of the accuracy function for ID3 on the Adult dataset by using a SGA as meta optimizer with a 70%-30% (Learning - Test) data partition.

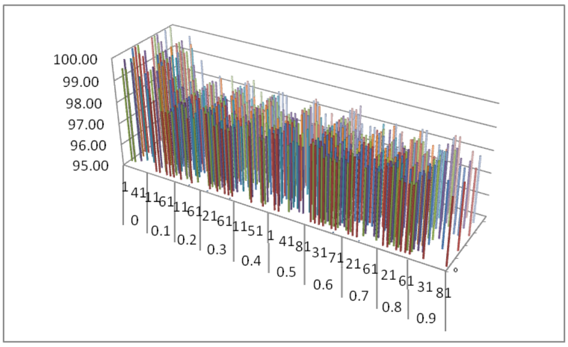

Figure 7: Exploration of the accuracy function for ID3 on the Mushroom dataset by using a SGA as meta optimizer with a 70%-30% (Learning - Test) data partition.

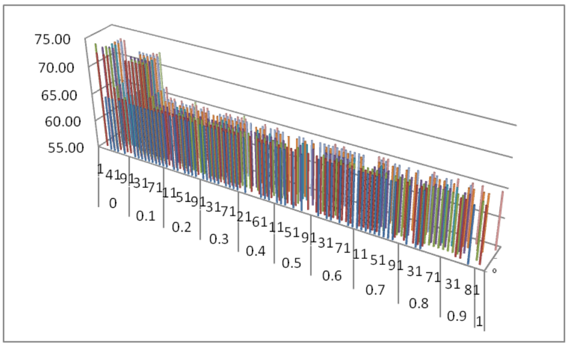

Figure 8: Exploration of the accuracy function for ID3 on the Diabetes dataset by using a SGA as meta optimizer with a 70%-30% (Learning - Test) data partition.

ter yielded the higher performance parameter sets. Within this subgroup of parameter sets, a minor variation was registered across the values for the Max Tree Depth parameter but is not clearly visible in the chart. A set of local maxima was registered in the values of 2 to 4 for Max Tree Depth. The minimum number of examples does not seem to have an effect across the range chosen.

The resulting accuracy landscape shows a maximum around the low values for Minimum Gain for Split and Maximum Depth followed by a long plain in the regions of higher values for Minimum Gain for Split.

The same Part 1 experiment for the Mushroom dataset showed a different behaviour, refer fig. 4. The highest value of accuracy corresponded to low values (0.0-0.2) of the Minimum Gain for Split parameter similar to the Adult data set. The resulting landscape appears to be rugged with stepped decrease from low to high Minimum Gain for Split values.

The accuracy is also higher for the lower values of the Minimum Examples for Split parameter and the accuracy does not vary much with changes in the Maximum Depth parameter except for a low at value 0.

The Diabetes experiments showed that low values (0.0-0.2) of the Minimum Gain for Split parameter yielded the higher performance parameter sets as in the Adult dataset. Within this subgroup of parameter sets, there is a variation across the values for the Max Tree Depth parameter. A set of local maxima was registered in the 2 - 4 values. The minimum number of examples does not seem to have an effect across the range chosen.

The resulting accuracy landscape shows a maximum around the low values for Minimum Gain for Split and Maximum Depth followed by a long plain in the regions of higher values for Minimum Gain for Split.

The maxima discovered using the larger training set to train set ratios similar but tend to be marginally higher on average thus supporting the hypothesis that more information available in the training set the better will be the accuracy measured over the test set. The difference in accuracy behaviour between the datasets is similar across experiments.

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 [2, 29]. Thus we want to explore how much a simple heuristic like a SGA can improve the search of the parameter space over the grid search heuristic.

The reported results have been obtained running the SGA in the same experimental setups as those described for the grid search in the previous section and on the two datasets. The SGA was run for both datasets 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, stopping rate 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. Each parameter set that was generated was used to generate and test ID3 trees using the same 10 random sample partitions of train and test data.

Fig. 6, fig. 7 and fig. 8 show the typical landscape (pattern of exploration) of the SGA the results of the exploration of the parameter space of the Adult, Mushroom and Diabetes using the 70%-30% split. At a first glance we can observe that in all three datasets, 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. Although the SGA executed more operations than the Grid Search due to the stopping rate of 100 and population of 40, it was, as we hoped, efficient in exploring the parameter space, discovering 'near' optimal parameter sets in the early epochs. this means that lower stopping rates can be set resulting in significant savings of computational power.

As can be seen from the results in Table 1 the SGA discovered the same regions of optimal parameter space as the Grid Search for all of the experiments. It is to be noted that for the Mushroom and Diabetes experiments these results were obtained in the earlier cycles of the evolutionary search. However in the Adult dataset experiments the optimal parameter sets

Table 1: Maximum Parameter Set Accuracy by Grid Search and SGA Search comparison

Dataset	Optimizer	30%/70% split	50%/50% split	70%/30% split
Adult	Grid Search	83.72%	83.99%	84.34%
	SGA	83.72%	83.99%	84.34%
Mushroom	Grid Search	99.89%	99.99%	100%
	SGA	99.89%	99.99%	100%
Diabetes	Grid Search	72.77%	72.58%	73.51%
	SGA	72.77%	72.58%	73.51%

Table 2: Maximum Parameter Set Accuracy per Epoch using SGA search with different Crossover rates

Dataset	Crossover rate	30%/70% split	50%/50% split	70%/30% split
Mushroom	25%	99.89%	99.99%	100%
	35%	99.89%	99.99%	100%
	50%	99.89%	99.99%	100%
	70%	99.89%	99.99%	100%
Diabetes	25%	72.77%	72.58%	73.51%
	35%	72.77%	72.58%	73.51%
	50%	72.77%	72.58%	73.51%
	70%	72.77%	72.58%	73.51%

were discovered at later epochs, i.e. epochs 76, 15 and 55 for the three train/test set ratios respectively.

8 Part 2: Effect of varying Crossover rate on SGA Performance

The objective for this part of the experimentation was to examine the effect of changing the crossover rate had on the performance of the SGA on the Mushroom and Diabetes datasets. The SGA was run for both datasets with crossover rates of 25%, 35%, 50% and 70%, whilst other parameters were kept as in the Part 1 experiments. Table 2 shows that the same maximum accuracy was discovered by the SGA with same train/test partitions but different Crossover rates e.g. for the Mushroom dataset with the 30% training/ 70% test partition the maximal accuracy discovered was 99.89% for all four Crossover rates. This was the same as the maximal accuracy discovered by Grid Search.

9 Comparison with other classifiers

Table 3 shows the accuracy results of classification by other studies on the same datasets as the ones used for this study.

The UCI library documentation accompanying the Adult data reported the accuracy values in Table 3 which were obtained after the removal of unknowns from the train/test sets. The NBTree shows the highest performance with 85.90% +-0.28 compared with the 84.34% with the 70%/30% train test split discovered by SGA meta-optimizer for the ID3 used in this study.

The documentation accompanying the Mushroom dataset reported a set of logical rules developed for benchmark purposes which gave an accuracy of 99.41%. The best accuracy obtained by this study using the SGA on the same dataset was 100% with the 70%/30% train/test split.

In [30] a number of classifiers including Discrim, C4.5,k-NN and Naivebay were applied to the Diabetes dataset using 12-fold cross-validation. The error rates reported were 0.225, 0.27, 0.324 and 0.262 respectively. These results are equivalent to an accuracy of 77.5%, 73% , 67.6% and 73.8% respectively as listed in Table 3. The best accuracy obtained by this study using the SGA on the same dataset was 73.51% using the 70%/30% training /test split compared with the highest obtained by Discrim at 77.5% accuracy.

10 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 has also been used to solve the problem of finding the optimal parameter sets and points to an efficient and effective alternative.

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

Dataset	SGA optimized ID3	Other	
	with 70%/30% split	Classifier	Accuracy
Adult	84.34%	C4.5	84.46% +-0.30
		Naive-Bayes	83.88% +-0.30
		NBTree	85.90% +-0.28
Mushroom	100%	Benchmark	99.41%
Diabetes	73.51%	Discrim	77.5%
		C4.5	73%
		k-NN	67.6%
		Naivebay	73.8%

The differences in the ID3 Accuracy landscapes for the Adult, Mushroom and Diabetes datasets evidences that for different datasets the optimal ID3 parameter sets can be found in regions of the parameter space which are particular to each dataset. This would imply that a single optimal parameter set should not be generalized for different datasets.

The number of datasets and algorithms was however limited so far and can only be used to draw conclusions of an indicative nature. In order to draw more robust inferences the behaviour and performance of the SGA metaoptimizer has to be examined across a wider base of algorithms and datasets in order to draw out any differences and problematics related to algorithm approach and dataset characteristics. Moroever large parameter spaces provide harder challenges to the search problems and so more work should be directed in this area in order to evaluate the SGA's potential in this regard.

Another limitation is that the stochastic element of the SGA leads to different performance results with every run, due to different initial populations and other random effects in the crossover and mutation operations. Thus multiple SGA experiments with the same conditions should be run in order to be able to generalise the SGA's performance on a particular algorithm-dataset experiment.

The results however still show 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.

11 Future Work

Our forthcoming experimental research efforts will be directed towards testing the robustness of the SGA based metaoptimization approach over wider variety of machine learner algorithms and datasets in order to evaluate its general applicability. We also intend to apply more rigorous statistical evaluation of the performances of both the parameter set configurations of the machine learner algorithms and the metaoptimizer itself. It is also planned to study of the effect of changing SGA parameters on the wider experimental base.

References:

[1] C. Blum and A. Roli, Metaheuristics in combinatorial optimization: Overview and conceptual comparison, *ACM Computing Surveys*, 2003, 35(3),pp. 268–308.

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

[3] K. A. Smith-Miles, Cross-disciplinary perspectives on meta-learning for algorithm selection, *ACM Computing Surveys*, 41(1),2009,pp. 6:1-6:25.

[4] M. Reif, F. Shafait, and A. Dengel, Meta-learning for evolutionary parameter optimization of classifiers, *Machine Learning*, 87(3), 2012,pp. 357–380.

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

[6] J. J. Grefenstette, Optimization of control parameters for genetic algorithms, *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1),1986,pp. 122 128.

[7] A.S. Staines,F. Neri,A Matrix Transition Oriented Net for Modeling Distributed Complex Computer and Communication Systems, *WSEAS Transactions on Systems*, 13, WSEAS Press (Athens, Greece),2014,pp. 12–22.

[8] 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, Springer, 2012,pp. 86–97.

[9] F. Hutter, H. H. Hoos, K. Leyton-Brown, and T. Stuutzle, Paramils:an automatic algorithm configuration framework, *Journal of Artificial Intelligence Research (JAIR)*, 36,2009,pp. 267–306.

[10] D. Wolpert, W.G. Macready,No Free Lunch Theorems for Optimization, *IEEE Transactions on Evolutionary Computation, April 1997*,1(1),1997,pp. 67–82.

[11] M. Camilleri, F. Neri, and M. Papoutsidakis, An Algorithmic Approach to Parameter Selection in Machine Learning using Meta-Optimization Techniques, *WSEAS Transactions on Systems*,WSEAS Press (Athens, Greece), 13,2014,pp. 202–213.

[12] F. Neri,M. Camilleri, An Algorithmic Approach to Parameter Selection in Machine Learning using Meta-Optimization Techniques, *Proceedings of the 5th International Conference on Circuits, Systems, Control, Signals (CSCS '14) and the 2nd International Conference on Acoustics, Speech and Audio Processing (ASAP '14), Salerno, Italy*, 2014, Editors F. Neri C. Guarnaccia, WSEAS Press(Athens, Greece),2014

[13] M. Papoutsidakis,D. Piromalis,F. Neri,M. Camilleri,Intelligent Algorithms Based on Data Processing for Modular Robotic Vehicles Control, *WSEAS Transactions on Systems*, 13, WSEAS Press (Athens, Greece),2014,pp. 242–251.

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

[15] H.B. Alwan,K.R. Ku-Mahamud,Solving SVM Model Selection Problem using ACO_R and $IACO_R$, *WSEAS Transactions on Computers*,WSEAS Press (Athens,Greece), 13,2014,pp. 277–288.

[16] M.A. Ahandani,M.T.V. Baghmisheh,Hybridizing Genetic Algorithms and Particle Swarm Optimization Transplanted into a Hyper-Heuristic System for Solving University Course Timetabling Problem, *WSEAS Transactions on Computers*,WSEAS Press (Athens,Greece),3(13),2013,pp. 128–143.

[17] J. Magalhaes-Mendes,A Comparative Study of Crossover Operators for Genetic Algorithms to Solve the Job Shop Scheduling Problem, *WSEAS Transactions on Computers*,WSEAS Press (Athens,Greece), 4(12),2013,pp. 164–173.

[18] A. E. Eiben, R. Hinterding, and Z. Michalewicz, Parameter control in evolutionary algorithms, *IEEE Transactions on Evolutionary Computation*, 3(2),1999,pp. 124–141.

[19] A. E. Eiben and S. K. Smit, Parameter tuning for configuring and analyzing evolutionary algorithms, *Swarm and Evolutionary Computation*, 1(1), 2011,pp. 19–31.

[20] M. Birattari,T. Stutzle,L. Paquete,K. Varrentrapp,A Racing Algorithm for Configuring Metaheuristics, *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, Morgan Kaufmann Publishers,2002,pp. 11–18.

[21] O. Maron,A.W. Moore,The Racing Algorithm: Model Selection for Lazy Learners, *Artificial Intelligence Review*,(11),1997,pp. 193–225.

[22] P. Balaprakash,M. Birattari,T. Stutzle,M. Dorigo,Improvement strategies for the F-race algorithm: Sampling design and iterative refinement, *In T Bartz-Beielstein, M Blesa, C Blum, B Naujoks, A Roli, G Rudolph, and M Sampels, editors, 4th International Workshop on Hybrid Metaheuristics, Proceedings, HM 2007,4771 of Lecture Notes in Computer Science*, Springer Verlag, Berlin, Germany,2007,pp. 108-122.

[23] M. Lopez-Ibanez, J. Dubois-Lacoste,T. Stutzle,M. Birattari,The Irace package, Iterated Race for Automatic Algorithm Configuration, *IRIDIA, Université Libre de Bruxelles, Belgium*, TR/IRIDIA/2011-004, url= http://iridia.ulb.ac.be/IridiaTrSeries/IridiaTr2011-004.pdf, 2011.

[24] J. Bergstra,Y. Bengio,Random Search for Hyper-Parameter Optimization, *Journal of Machine Learning Research*, 13,(feb),2012,pp. 281-305.

[25] C. Neumiller, S. Wagner, G. Kronberger, and M. Affenzeller, Parameter meta-optimization of metaheuristic optimization algorithms, In Moreno-Daz, R., Pichler, F. & Quesada-Arencibia, A. (Eds.), *Computer Aided Systems Theory EUROCAST 2011*, Vol. 6927 of *Lecture Notes in Computer Science* Springer Berlin Heidelberg, 2012,pp. 367–374.

[26] A. Kumar,S. Kumar,Decision Tree Based Learning Approach for Identification of Operating System Processes, *WSEAS Transactions on Computers*,WSEAS Press (Athens,Greece), 13,2014,pp. 277–288.

[27] A. Frank and A. Asuncion, UCI Machine Learning Repository, http://archive.ics.uci.edu/ml, 2010.

[28] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, The weka data mining software: An update, *SIGKDD Explorations*, 11(1), 2009.

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

[30] D. Michie, D. Spiegelhalter, and C. Taylor, Machine learning, neural and statistical classification, New York:Ellis Horwood,1994.