An Algorithmic Approach to Parameter Selection in Machine Learning using Meta-Optimization Techniques

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

FILIPPO NERI
University of Naples
Dept. of Computer Science
Naples
Italy
nerifil@gmail.com

MICHALIS PAPOUTSIDAKIS
Department of Automation
Technological Institute of Piraeus
P.Ralli & Thivon 250, 12244, Athens
GREECE
mipapou@teipir.gr

Abstract: The process of identifying the optimal parameters for an optimization algorithm or a machine learning one is usually costly, involves the search of a large, possibly infinite, space of candidate parameter sets, and may not guarantee optimality. Various attempts have been made to automate this process. 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 a selected dataset.

Key–Words: metaheuristics optimization , machine learning

1 Introduction

This study evidences the necessity for academic papers 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 [1, 5, 23, 21, 19, 6].

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 [8] 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) [5], used as a meta-optimizer, in finding good parameters for an ID3 like decision tree learner [20].

The long term goals of our research 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).

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.

We also believe that the meta optimization framework that we are presenting here could also be applied to a variety of learning approaches and real world problems such as for instance in [9, 22, 14, 13, 18, 12, 15, 16, 17]. Finally we like to remind a number of approaches that could benefit under several forms of our current work [24, 27, 26, 25, 28, 32, 33, 34, 37, 35, 36, 38, 40, 39, 29, 30, 31].

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 [3, 2] 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.
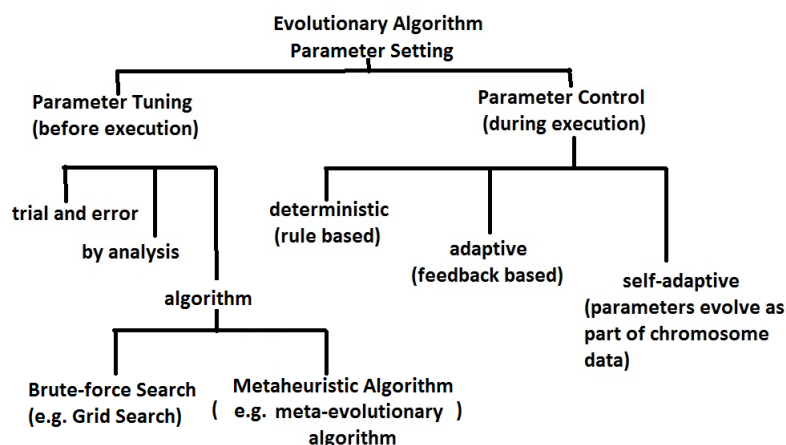
Figure 1: Taxonomy of meta optimization methods

Researchers have tried to classify research studies in meta optimization of learning parameters using a taxonomy [3] like the one in Fig. 1 [3] 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 [21]. We, on the other hand, believe that each datset requires specific parameter optimization for a given learning systems. Also the work in [21] is impractical as it would require the existence of a database of several optimized $< datasets, parameter settings >$ pairs to allow the case based reasoning approach 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 used 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 2, a graphical representation of the meta-optimization task is reported. In the paper, L1 will be a decision tree learner and L2 will be a simple genetic algorithm (SGA). The relationships among meta-level optimization, the meta-optimization problem and the base learning algorithm are illustrated in fig. 2.

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 used was based on ID3[20] 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 used in the metaoptimizer system was a simple Genetic Algorithm (SGA). 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
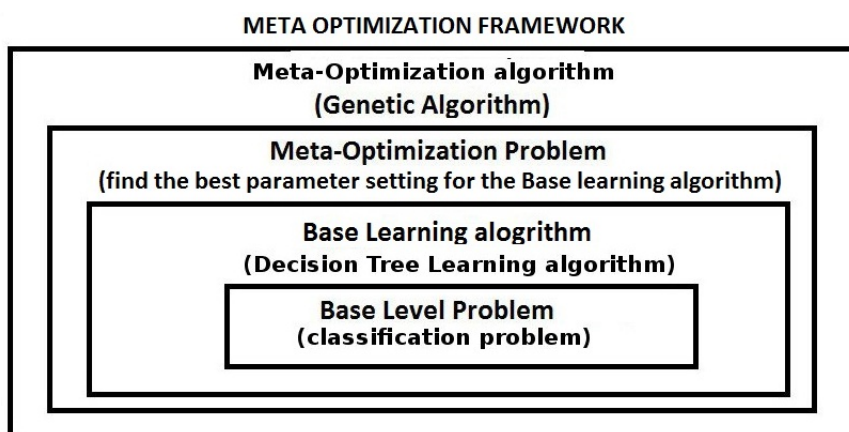
Figure 2: The meta-optimization framework.

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.

The experimental system set up for this study is thus composed of three component subsystems: the meta-optimizer, the evaluator, the machine learner. The meta-optimizer subsystem is the controlling subsystem whose objective is to discover the optimal or near optimal parameter set for the machine learner. It embodies the SGA (the meta-optimizer algorithm) which guides and controls the search of the machine learner's parameter space. Each parameter set that it discovers is passed to the machine learner evaluator subsystem. This evaluator subsystem then calls the machine learner subsystem on the 10 different training/test set partitions of the same dataset for the statistical validation.

The machine learner subsystem learns a classifier tree on the given training set partition, tests it on the related test data partition and returns the performance of the parameter set in terms of percentage accuracy of correctly classified instances. The evaluator then averages the parameter set performance across the 10 machine learner runs and returns this average performance to the meta-optimizer subsystem. The meta-optimizer then collates the results of the set of parameter sets, represented by the population of chromosomes and determines the composition of the next generation through the processes of elite injection, se-

lection, crossover and mutation.

The experimental study was carried out in two parts:

The first part was a systematic exploration of two test datasets using a Grid Search algorithm to explore the ID3 accuracy landscape of the two datasets.

A broad range of ID3 parameter set values were applied to the Grid Search based system 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 based system to examine the ability of the SGA to explore the same parameter 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. In this part of the study the same test datasets were used with Crossover rates of 25%, 35%, 50% and 70%.

The first dataset, *Credit Card Approval* (CCA) , was obtained from the dataset library of the WEKA Data Mining Software [7] which was originally sourced from the UCI repository [4] and is available under the name of Credit Approval. The CCA dataset contained 690 records of persons. Each instance was 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 was to learn from the available data when to classify an unseen credit card applicant as approved or not.

The second dataset was an artificial dataset *Artificial ILPD* (aILPD), which was originally derived from the *Indian Liver Patients* (ILPD) dataset (accessible from the UCI repository [4]). The data and the class mix in aILPD datset were different from the original

ILPD dataset. The aILPD dataset contained 583 instances with 10 attributes and a class attribute. The classification problem in this case was similar to that of the CCA dataset, i.e. to correctly classify unseen test instances.

4 Results of the Experiments: 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 in the parameter search system) 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 15 (for the CCA dataset and from 0 to 10 for the aILPD 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 Split was set from 1 to 101 in steps of 10. These settings resulted in a uniform point cover of 1936 and 1331 different parameter sets for the CCA and aILPD 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 58080 and 39930 ID3 evaluations for the CCA and aILPD datasets respectively.

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

In this section we study the accuracy landscape produced by a parameter search system based on 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

system, secondly we wanted to illustrate the 'ruggedness' of the accuracy function produced by the space of parameter sets input into the decision tree learner.
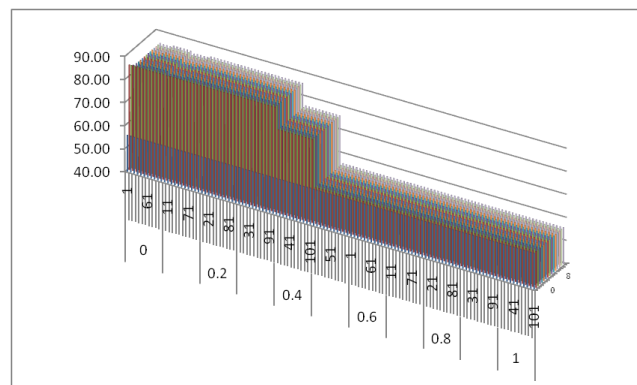

Figure 4: Accuracy function over the parameter space with a 70%-30% (Learning - Test) split of the CCA dataset obtained by using a grid search meta optimizer.The vertical axis report the accuracy value, whilst the left horizontal axis reports two parameter ranges, the Minimum Gain for Split (left outer)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

In figures 3 and 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 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 fig. 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 fig. 4.
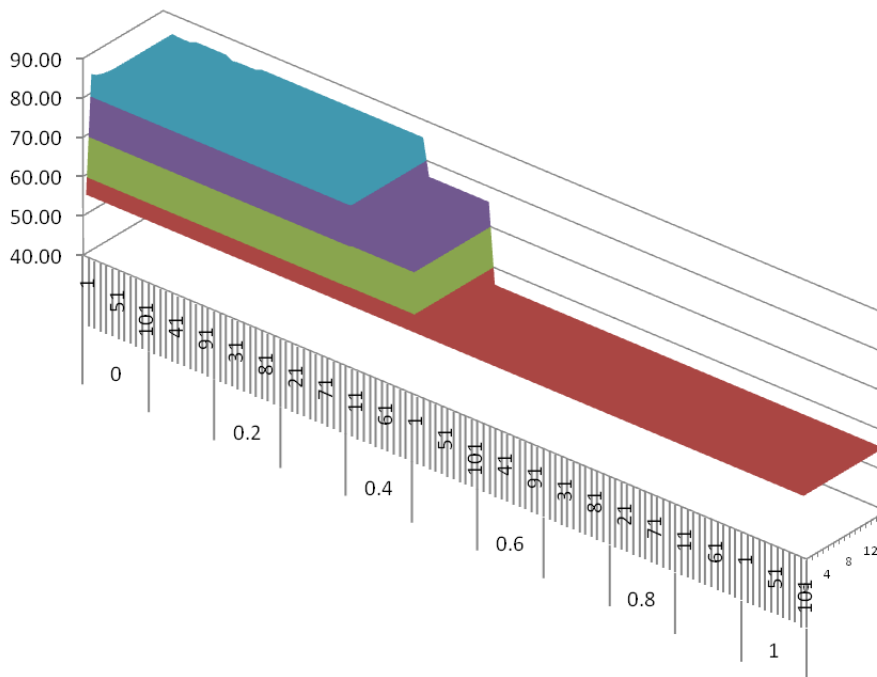
Figure 3: Accuracy function over the parameter space with a 70%-30% (Learning - Test) split of the CCA dataset obtained by using a grid search meta optimizer.The vertical axis report the accuracy value, whilst the left horizontal axis reports two parameter ranges, the Minimum Gain for Split (left outer) and the Minimum Examples for Split (left inner). The right horizontal axis represents the Maximum Tree Depth ID3 was allowed to grow.

The results of the CCA experiments for the 70%/30% train/test partition, as illustrated in figures 3 and 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.
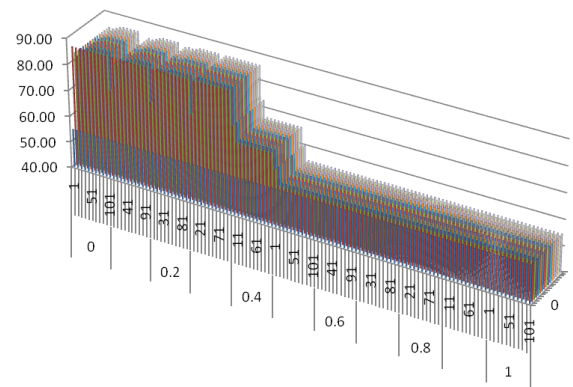

Figure 6: Accuracy function over the parameter space with a 30%-30% (Learning - Test) split of the CCA dataset obtained by using a grid search meta optimizer.
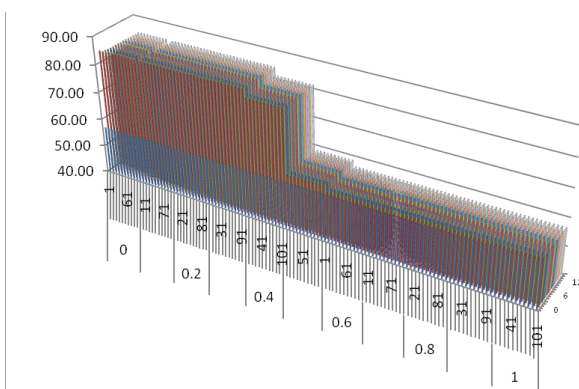

Figure 5: Accuracy function over the parameter space with a 50%-50% (Learning - Test) split of the CCA dataset obtained by using a grid search meta optimizer.

The results of the aILPD experiments for the 70%/30% train/test partition, as illustrated in fig. 7, showed that the region of highest values of accuracy corresponded to the 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 (those close to value 1) gave the highest accuracy (unlike CCA). All the other regions of parameter space exhibited continuous planes
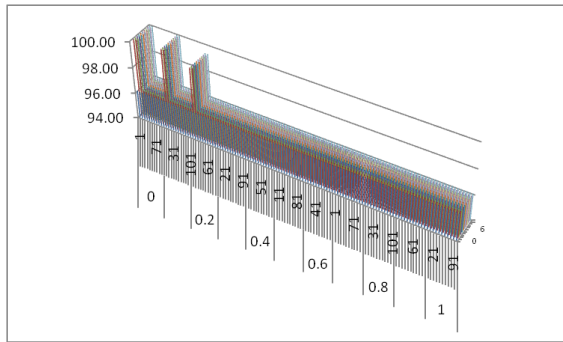
Figure 7: Accuracy function over the parameter space with a 70%-30% (Learning - Test) split of the aILPD dataset obtained by using a grid search meta optimizer.

Figure 8: Accuracy function over the parameter space with a 50%-50% (Learning - Test) split of the aILPD dataset obtained by using a grid search meta optimizer.
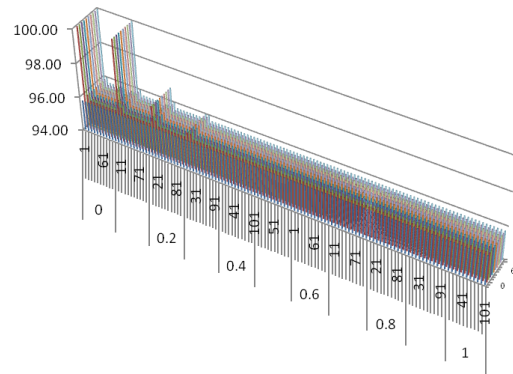

Figure 9: Accuracy function over the parameter space with a 30%-70% (Learning - Test) split of the aILPD dataset obtained by using a grid search meta optimizer.
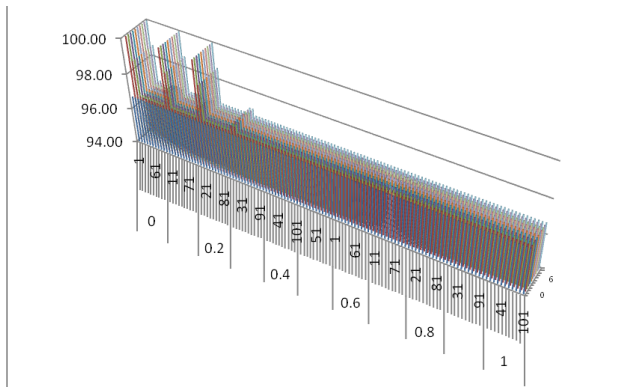
Optimizer	Partition	Value	Epoch
Grid Search	30%/70%	85.96%	
	50%/50%	84.35%	
	70%/30%	85.51%	
SGA	30%/70%	85.96%	1
	50%/50%	84.35%	1
	70%/30%	85.51%	1

Table 1: CCA Maximum Parameter Set Accuracy

of lower accuracy. The resulting landscape appears initially hilly in the low parameter values with a long plain following in the higher value regions. A further observation is that the 'hills' are highest in the region of lower values for Minimum Gain. Increasing Minimum tree depth did not appear to affect accuracy.

Figure 4 and fig. 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 fig. 5 and fig. 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. Figure 5 shows a slightly more stepped progression of accuracy as the value of Mininum Gain for Split increases. Fig. 6 shows even more variation across the landscape with decreases in accuracy on the lower

and higher values of Minimum Number of Examples for Split rendering the landscape more 'hilly' in the regions of smaller values for Minumum Gain for Split.

Similarily fig. 8 and fig. 9 showing the aILPD Accuracy Landscape on smaller train/test partitions show an overall similarity with the 70%/30% partition experiment. Again, small differences can be noted. Figure 8 shows a slightly more stepped progression of accuracy within the regions of lower values for Mininum Gain for Split and Minimum Number of Examples for Split. Fig. 9, the third and fourth region of higher accuracy, corresponding to Min Gain of 0.2 and 0.3 are much reduced when compared to the previous two aILPD experiments.

This evidences that fact that the choice of training/test partition size of the same dataset and the same algorithm can also can give different results (see Table 1) and overall performance over the same parameter space.

4.2 Experiments using 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 [5, 10]. Thus we

Optimizer	Partition	Value	Epoch
Grid Search	30%/70%	100%	
	50%/50%	100%	
	70%/30%	100%	
SGA	30%/70%	100%	27
	50%/50%	100%	2
	70%/30%	100%	1

Table 2: aILPD Maximum Parameter Set Accuracy

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 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, 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. The ID3 parameter set generated by each individual of the population was then used to generate and test ID3 trees using the same 10 random sample partitions of train and test data as used in the grid search experiments.

Figures 10 and 11 show the 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.

Tables 1 and 2 show the maximum ID3 accuracy evaluated in each experiment. The SGA discovered the same maximal values for the same train/test splits for both datasets. Interestingly the maximal values discovered by the SGA were in the first epoch in both cases. This early discovery of the maximal value may be more attributable to the size and randomness of the initial populations and the nature of the datasets themselves rather than the evolutionary approach of the meta-optimizer.

5   Comparison with other classifiers

In [11] 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
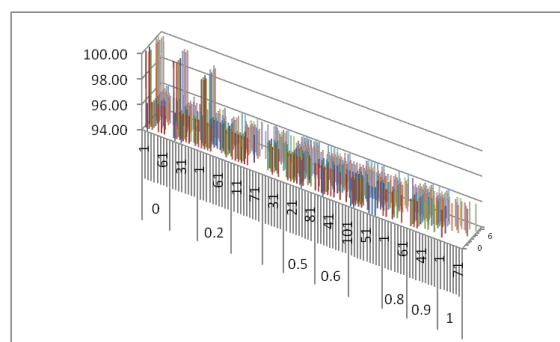

Figure 11: Accuracy function over the parameter space with a 70%-30% (Learning - Test) split of the aILPD dataset obtained by using the SGA meta optimizer.

0.151 respectively. These results are equivalent to an accuracy of 86.9%, 84.5% 81.9% and 84.9% respectively. 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.

An experiment was run for comparison purposes using the Random Forest and SimpleCart classifiers in the WEKA library [7] on the same aILPD 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 aILPD dataset. The best result achieved by the SGA experiments using ID3 for the aILPD data set was also 100/% for the three training /test partitions.

6   Conclusion

In the paper, we have compared Grid Search and SGA as meta optimizing systems 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 system to provide coarse but uniform exploration of the parameter space. The heuristic based system utilising the SGA instead has been used to solve in an efficient and effective way the problem to find the optimal parameter sets.

The results show that researchers in machine learning systems 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.
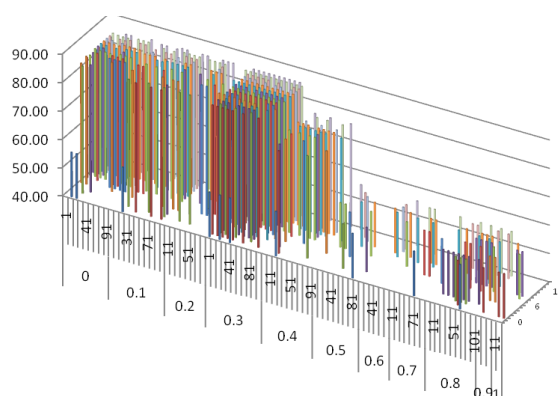
Figure 10: Accuracy function over the parameter space with a 70%-30% (Learning - Test) split of the CCA dataset obtained by using the SGA meta optimizer.

Dataset	Crossover rate	30%/70% split		50%/50% split		70%/30% split	
		Accuracy	Epoch	Accuracy	Epoch	Accuracy	Epoch
CCA	25%	85.96%	1	84.35%	1	85.51%	1
	35%	85.96%	1	84.35%	1	85.51%	1
	50%	85.96%	1	84.35%	1	85.51%	1
	70%	85.96%	1	84.35%	1	85.51%	1
aILPD	25%	100%	27	100%	2	100%	1
	35%	100%	6	100%	1	100%	1
	50%	100%	48	100%	3	100%	1
	70%	100%	8	100%	13	100%	1

Table 3: Maximum Parameter Set Accuracy per Epoch using SGA search with different Crossover rates

References:

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

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

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

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

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

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

[7] 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.

[8] 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.

[9] A. M. Kassem, Fuzzy-Logic Based Self-tuning PI Controller for High-Performance Vector Controlled Induction Motor Fed by PV-Generator,*WSEAS Transaction on Systems,WSEAS Press*, 12(1):22–31, 2013.

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

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

[12] F. Neri, Traffic packet based intrusion detection: decision trees and genetic based learning evaluation", *WSEAS Transactions on Computers*, 9(4): 1017-1024, 2005.

[13] F. Neri, Learning and Predicting Financial Time Series by Combining Evolutionary Computation and Agent Simulation, *Applications of Evo-*

lutionary Computation, EvoApplications, LNCS 6625,Springer,Heidelberg, pages 111–119, 2011.

[14] F. Neri, PIRR: a Methodology for Distributed Network Management in Mobile Networks, *WSEAS Transactions on Information Science and Applications, WSEAS Press*, 5(3):306–311,2008.

[15] A. S. Staines, and F. Neri, A Matrix Transition Oriented Net for Modeling Distributed Complex Computer and Communication Systems, *WSEAS Transactions on Systems*, in press, 2014.

[16] F. Neri, Software agents as a versatile simulation tool to model complex systems, *WSEAS Transactions on Information Science and Applications*, issue 5, vol. 7, pp. 609-618, 2010.

[17] F. Neri, Empirical investigation of word-of-mouth phenomena in markets: a software agent approach, *WSEAS Transactions on Computers*, issue 8, vol. 4, pp. 987-994, 2015.

[18] F. Neri, Quantitative Estimation of Market Sentiment: a discussion of two alternatives", *WSEAS Transactions on Systems,WSEAS Press*, 11(12):691–702,2012.

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

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

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

[22] A. M. Reza, Adaptive Noise Filtering of Image Sequences in Real Time, *WSEAS Transaction on Systems,WSEAS Press*, 12(4):189–201,2013.

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

[24] M. Papoutsidakis, F. Neri, and M. Camilleri, Intelligent Algorithms Based on Data Processing for Modular Robotic Vehicles Control, *WSEAS Transactions on Systems*, in press, 2014

[25] P. Hajek, and F. Neri, An introduction to the special issue on computational techniques for trading systems, time series forecasting, stock market modeling, financial assets modeling *WSEAS Transactions on Business and Economics* 10(4): 201–292, 2013.

[26] M. Azzouzi, and F. Neri, An introduction to the special issue on advanced control of energy systems (2013) *WSEAS Transactions on Power Systems*, 8(3): 103, 2013.

[27] Z. Bojkovic, and F. Neri, An introduction to the special issue on advances on interactive multimedia systems *WSEAS Transactions on Systems*, 12(7): 337–338, 2013.

[28] L. Pekar, and F. Neri, An introduction to the special issue on advanced control methods: Theory and application *WSEAS Transactions on Systems*, 12(6): 301–303, 2013.

[29] C. Guarnaccia, and F. Neri, An introduction to the special issue on recent methods on physical polluting agents and environment modeling and simulation, *WSEAS Transactions on Systems*, 12(2): 53–54, 2013.

[30] F. Neri, An introduction to the special issue on computational techniques for trading systems, time series forecasting, stock market modeling, and financial assets modeling, *WSEAS Transactions on Systems*, 11(12): 659–660, 2012.

[31] M. Muntean, and F. Neri, Foreword to the special issue on collaborative systems, *WSEAS Transactions on Systems*, 11(11): 617, 2012.

[32] L. Pekar, and F. Neri, An introduction to the special issue on time delay systems: Modelling, identification, stability, control and applications, *WSEAS Transactions on Systems*, 11(10): 539–540, 2012.

[33] C. Volos, and F. Neri, An introduction to the special issue: Recent advances in defense systems: Applications, methodology, technology, *WSEAS Transactions on Systems*, 11(9): 477–478, 2012

[34] F. Neri, Open research issues on Computational Techniques for Financial Applications *WSEAS Transactions on Systems*, in press, 2014.

[35] C.F. de Melo, and F. Neri, Open research issues on Orbital Dynamics and Spacecraft Attitude Control, *WSEAS Transactions on Systems*, in press, 2014.

[36] I. Eleftheriadis, N. Loukeris, and F. Neri, Open research issues on Advanced Control Methods: Theory and Application, *WSEAS Transactions on Systems*, in press, 2014.

[37] M. Panoiu, and F. Neri, Open research issues on Modeling, Simulation and Optimization in Electrical Systems, *WSEAS Transactions on Systems*, in press, 2014.

[38] S. P. Karthikeyan, and F. Neri, Open research issues on Deregulated Electricity Market: Investigation and Solution Methodologies, *WSEAS Transactions on Systems*, in press, 2014.

[39] C. Ciufudean, and F. Neri, Open research issues on Multi-Models for Complex Technological Systems, *WSEAS Transactions on Systems*, in press, 2014.

[40] A. V. Doroshin, and F. Neri, Open research is-
sues on Nonlinear Dynamics, Dynamical Systems
and Processes, *WSEAS Transactions on Systems*, in
press, 2014.