An Evaluation of Over-Fit Control Strategies for Multi-Objective Evolutionary Optimization

Paulo V. W. Radtke, Tony Wong, *Member, IEEE*, and Robert Sabourin, *Member, IEEE*

***Abstract*— The optimization of classification systems is often confronted by the solution over-fit problem. Solution over-fit occurs when the optimized classifier memorizes the training data sets instead of producing a general model. This paper compares two validation strategies used to control the over-fit phenomenon in classifier optimization problems. Both strategies are implemented within the multi-objective NSGA-II and MOMA algorithms to optimize a Projection Distance classifier and a Multiple Layer Perceptron neural network classifier, in both single and ensemble of classifier configurations. Results indicated that the use of a validation stage during the optimization process is superior to validation performed after the optimization process.**

I. Introduction

One of the challenges in example-based learning is to avoid the over-fitting problem where the learner memorizes the training data set instead of finding hidden relations within the data. The over-fitting problem is well-known in the area of classifier training where the classifier parameters are adjusted based on the current accuracy to classify the training data set [1]. Learning over-fit often occurs after a number of training iterations and the classifier starts to memorize the training data instead of producing a more general model of the data. At this point, it is said that the classifier becomes over-fitted to its training data set. This phenomenon is illustrated in Fig. 1. On early iterations, the error rate of the classifier decreases on the training data set and on unknown observations. After iteration t_{stop}, the error rate on the training set keeps decreasing, but on the unknown observations the error rate starts to increase. This effect is due to the over-fit to the training data set.

Thus, the classifier training problem is to determine the iteration t_{stop} at which the training procedure must stop. The t_{stop} determination can be achieved through a validation strategy using a validation data set of observations unknown to the training procedure. At each training iteration, the classifier parameters are adjusted as usual and its accuracy is evaluated on the validation data set. In this way, t_{stop} is determined as the last training iteration during which the classifier improved its accuracy on the validation data set.

This paper extends the works in [2], [3], tackling the over-fit issue observed during the optimization of classification systems. The novelty is the proposal of an efficient strategy to control solution over-fit in the context of multi-objective optimization for example based learning, and a comparison with traditional methods. Strategies are tested on two optimization algorithms, the *Fast Elitist Non-Dominated Sorting Algorithm* (NSGA-II) [4] and the *Multi-Objective Memetic Algorithm* (MOGA), introduced by the authors in [2]. MOMA also differs from traditional MOGAs (multi-objective genetic algorithms) by replacing the Pareto-based approach to guide optimization by a concept referred to as the *decision frontier*, adapted to the optimization of learner algorithms. The decision frontier requires two objective functions, one integer, used to define the optimization slots, and a real valued objective function optimized for each possible optimization slot. A second difference is the use of an auxiliary archive to help overcome solution over-fit, which is extended in this paper to effectively control the over-fit.

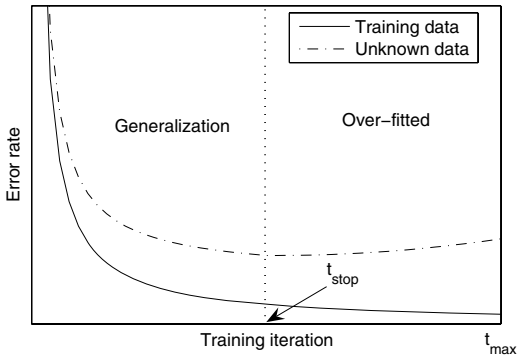


Fig. 1. Classifier training error rate curves.

The outline of this paper is as follows: Section II explains the over-fit effects encountered in multi-objective optimization of classifier systems. Section III presents the over-fit control strategies. Section IV covers the experimental protocol using in the comparison of the over-fit control strategies and the obtained results. Finally, conclusions are drawn in Section V.

II. Over-Fit Effects in Multi-Objective Optimization

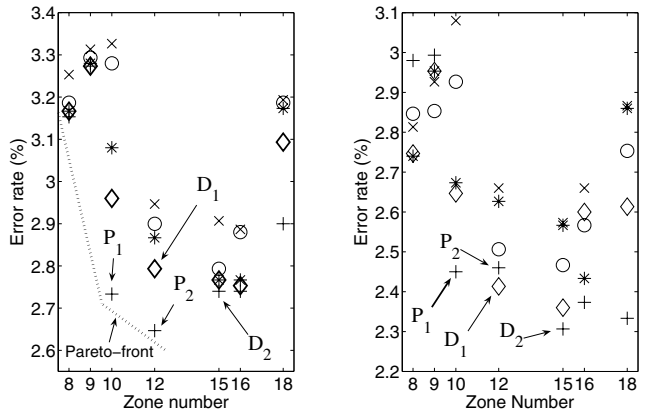
Solution over-fit also occurs when classification systems are optimized using evolutionary algorithms in a wrapper approach. It is because the optimization process becomes a learning process and the search for solutions is based on the wrapped classifier accuracy that is computed using

Paulo V. W. Radkte, T. Wong and R. Sabourin are with the Automated Manufacturing Engineering Department, Ècole de technologie supérieure, University of Québec, 1100 Notre-Dame West, Montral (Québec) Canada, H3C 1K3 (e-mail: radtke@livia.etsmtl.ca, tony.wong@etsmtl.ca, robert.sabourin@etsmtl.ca).

Paulo V. W. Radkte and R. Sabourin are also with the Pontifícia Universidade Católica do Paraná, R. Imaculada Conceição 1155, CEP 80215-901, Curitiba, PR, Brazil.

an *optimization* data set. Solutions found at the end of the optimization process might be over-fitted to the *optimization* data set. This effect is observed even when a validation procedure is used to train the wrapped classifier. It is thus necessary to install a validation procedure in the optimization process in order to select solutions with good generalization power.

As an example, Fig. 2 shows the over-fit phenomenon in the optimization of the intelligent feature extraction (IFE) process [2] associated to a PD (Projection Distance) classifier [5]. The set of solutions was generated by a MOGA. Figure 2.a is the optimization objective space associated with the *optimization* data set accuracy and is used to guide the optimization process. To verify a solution's generalization power, a set of unknown observations is used to evaluate the PD classifier's accuracy, producing another objective space called unknown observations objective space (shown in Fig. 2.b). The solution labeled P_2 is the solution with the smallest error rate obtained on the Pareto front in the optimization objective space. However, as can be seen in Fig. 2.b, solution P_2 does not generalize as well as the solution P_1 on the unknown data set. Furthermore, good solutions obtained during the optimization process perform differently on unknown observations. Again, solution P_2 in Fig. 2.a dominates solution D_1 in the optimization objective space but is dominated by solution D_1 in the unknown observations objective space. Clearly, solution P_2, even though found on the Pareto front of the optimization objective space, has less generalization power than another solution not actually on the Pareto front.



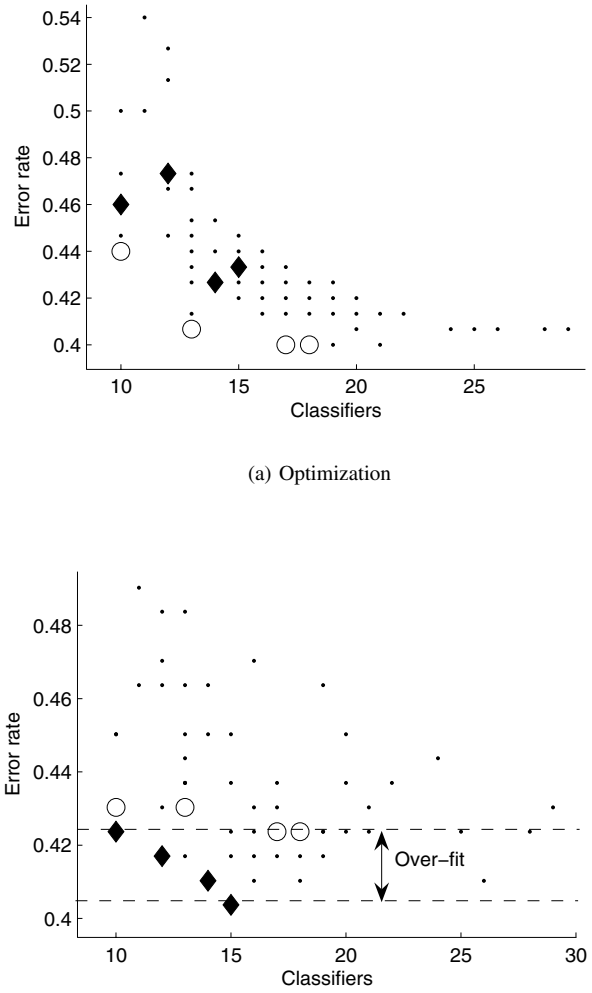
(a) *Optimization* data set (b) Unknown observations

Fig. 2. Objective spaces produced by a MOGA during the optimization of a feature extraction process.

III. Over-fit Control Strategies

One strategy used to overcome this type of over-fit following the optimization process is to validate final solutions on the Pareto front with yet another set of unknown observations – the *selection* data set. This over-fit control strategy is able to select P_1 in Fig. 2 as the most accurate generalization solution. It produces better results than selecting solutions based solely on the accuracy of the *optimization* data set alone [6]–[8]. In spite of this, the use of a *selection* data set can not prevent dominated solutions D_1 and D_2 which has great generalization power from being discarded by the MOGA.

The main shortcoming of this over-fit control strategy is that the solution validation is only performed once – after the optimization process has been completed. This deficiency is illustrated in Fig. 3 showing the solutions in the optimization objective space and the validation objective space generated at generation $t = 14$ by the NSGA-II optimizer for the EoC (Ensemble of Classifiers) [9], [10] optimization proposed in [3].



(a) Optimization

(b) Validation

Fig. 3. Actual solutions generated atgeneration $t = 14$ by the optimization and validation processes with NSGA-II for an EoC problem.

In Fig. 3, points are candidate solutions in the current generation, circles represent solutions in the current Pareto front, and diamonds are the solutions on the Pareto front in the validation objective space. From the Fig. 3, it is clear that solutions with good generalization power are often eliminated by genetic selection which emphasizes solutions with good performance on the *optimization* data set. Hence, it appears that the candidate solutions validation should be executed in all generations during the optimization process.

In the case of multi-objective evolutionary optimization, validating candidate solutions through generations can be accomplished by storing the validated solutions in an auxiliary archive. This approach is demonstrated through the generations in EoC optimization with both NSGA-II and MOMA in Figs. 4 and 5 respectively. The points represent the complete search space covered by the algorithms, and each point is an MLP EoC where individual classifiers were trained with single-split validation. Diamonds are EoCs belonging to the approximation set (Pareto front for NSGA-II and the decision frontiers for MOMA) at generation t. The first column (a) details algorithm convergence during the optimization process. In the second column (b), the same solutions are projected on the validation objective function space. Finally, the third column simulates convergence in the auxiliary archive obtained by validating the population at each generation t with the *selection* data set.

Both Figs. 4 and 5 confirm that there is still over-fit when solutions are only validated once the optimization process has been completed. For the NSGA-II in Fig. 4.b.3, the best candidate solution is 13.89% over-fitted when compared to the lowest error rate in the *selection* data set. The same is observed with MOMA in Fig. 5.b.3, where the best candidate solution is 9.75% over-fitted. The second column in both figures also indicates that some EoCs that perform well in the *selection* data set are discarded by both algorithms. These observations further confirm the need to validate solutions at each generation t to obtain the approximation sets in Figs. 4.c.3 and 5.c.3. This validation strategy to control solution over-fit is hereafter referred to as *global validation*.

The global validation strategy for MOGAs is given in Algorithm 1. A multi-objective genetic algorithm evolves the population P_t for a number of generations. At each generation, the population P_{t+1} is validated and the auxiliary archive S is updated with solutions that have good generalization power. Similar to the validation strategy used in training classifiers, the validation stage provides no feedback to the MOGA. At the end of the optimization process, the best candidate solutions are stored in the archive S and outperformed solutions are removed from S. The remainder of this section details the approach to adapt both NSGA-II and MOMA to the global validation strategy.

A. Adapting the Global Validation Strategy to NSGA-II

An empty auxiliary archive S is added in order to apply the proposed global validations strategy to the NSGA-II algorithm. Validated solutions are inserted in S at each generation according to the auxiliary archive update procedure.

Result: Auxiliary archive S
Creates initial population P_1 with m individuals;
$S = \emptyset$;
t=1;
while $t <$ *maximum iterations* **do**
  evolve P_t to P_{t+1};
  validate P_{t+1} with the selection data set;
  update the auxiliary archive S with individuals from P_{t+1} based on their fitness from the validation process;
  t=t+1;
end

Algorithm 1: Pseudo-code showing the proposed global validation strategy to control over-fit.

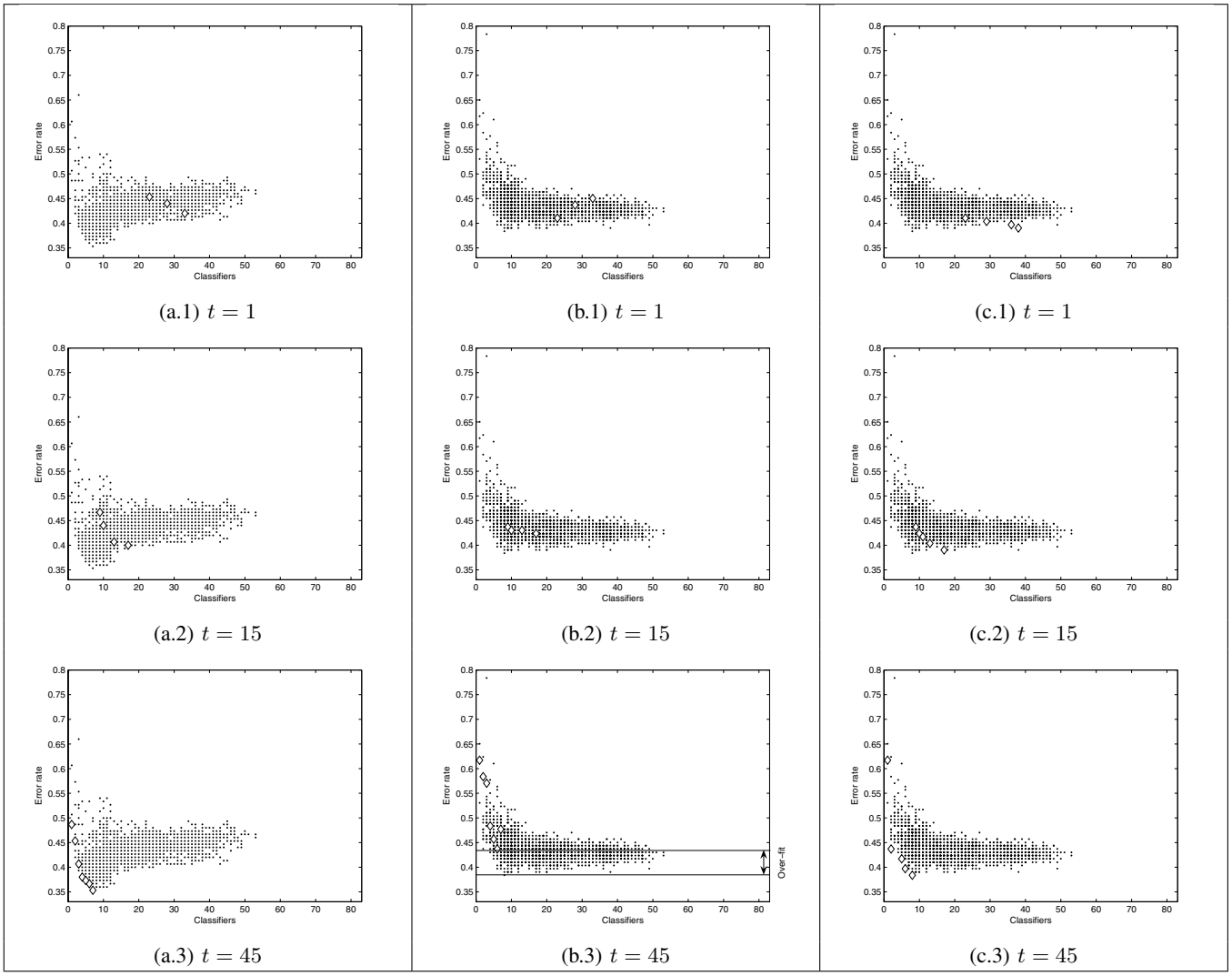
During this procedure, the *optimization* data set is replaced temporarily by the *selection* data set to evaluate optimization objective functions. Each solution x in the current population is tested for insertion into S. If x is inserted into S, solutions dominated by x in S are checked and eliminated accordingly. Pseudo-code in Fig. 5 and Fig. 6 shows this over-fit control strategy. Further details on NSGA-II can be found in [4].

Result: **The auxiliary archive S*
Creates initial population P_1 with m individuals;
*$S = \emptyset$;
t=1;
while $t <$ *maximum iterations* **do**
  $R_t = P_t \cup Q$;
  F=fast-non-dominated-sort(R_t);
  while $|P_{t+1}| + |F_i| \leq m$ **do**
    $P_{t+1} = P_{t+1} \cup F_i$;
    crowding-distance-assignment(F_i);
    $i = i + 1$;
  end
  Sort(F_i, $\prec_n$);
  $P_{t+1} = P_{t+1} \cup F_i[1 : (N - |P_{t+1}|)]$;
  Q_{t+1} = make-new-pop(P_{t+1});
  $t = t + 1$;
  **Update auxiliary archive S*;
end

Algorithm 2: Pseudo code showing the NSGA-II with global validation to control over-fit.

B. Adapting the Global Validation Strategy to MOMA

Adapting MOMA to the global validation strategy is a simpler process. It is sufficient to temporarily replace the *optimization* data set by the *selection* data set in the original procedure to update the auxiliary archive S defined in MOMA. The modified auxiliary archive update procedure is indicated in Algorithm 4. The detailed MOMA and related mathematical definitions are presented in [2].



(a.1) $t = 1$ (b.1) $t = 1$ (c.1) $t = 1$

(a.2) $t = 15$ (b.2) $t = 15$ (c.2) $t = 15$

(a.3) $t = 45$ (b.3) $t = 45$ (c.3) $t = 45$

Fig. 4. EoC optimization with NSGA-II at generation t: (a) Pareto front on the *optimization* data set; (b) Pareto front projected on the *selection* data set; and (c) the actual Pareto front in the *selection* data set. The most accurate solution in b.3 is over-fitted by 13.89% in comparison to the most accurate solution in Fig. c.3.

Data: Current population P_t and the auxiliary archive S
Result: The modified auxiliary archive S
Replaces optimization data set by the selection data set for objective function evaluation;
Calculate objective functions for all solutions in P_t;
forall $x^i \in P_t$ **do**
  if $\nexists x^j, x^j \in S \wedge x^j \succ x^i$ **then**
    $D = \{d \in S | x^i \succ d\}$;
    $S = S \backslash D \cup \{x^i\}$;
  end
end
Restores optimization data set for objective function evaluation;

Algorithm 3: Auxiliary archive update procedure for NSGA-II.

Data: Current population P_t and the auxiliary archive S
Result: The modified auxiliary archive S
Replaces optimization data set by the selection data set for objective function evaluation;
forall $x^i \in P_t$ **do**
  Determines the slot S^l solution x^i relates to;
  if $o_2(x^i) < o_2(W(S^l))$ *and* $x^i \notin S^l$ **then**
    $S^l = S^l \cup \{x^i\}$;
    if $|S^l| > max_{S^l}$ **then**
      $S^l = S^l \backslash \{W(S^l)\}$;
    end
  end
end
Restores optimization data set for objective function evaluation;

Algorithm 4: Modified auxiliary archive update procedure for MOMA.

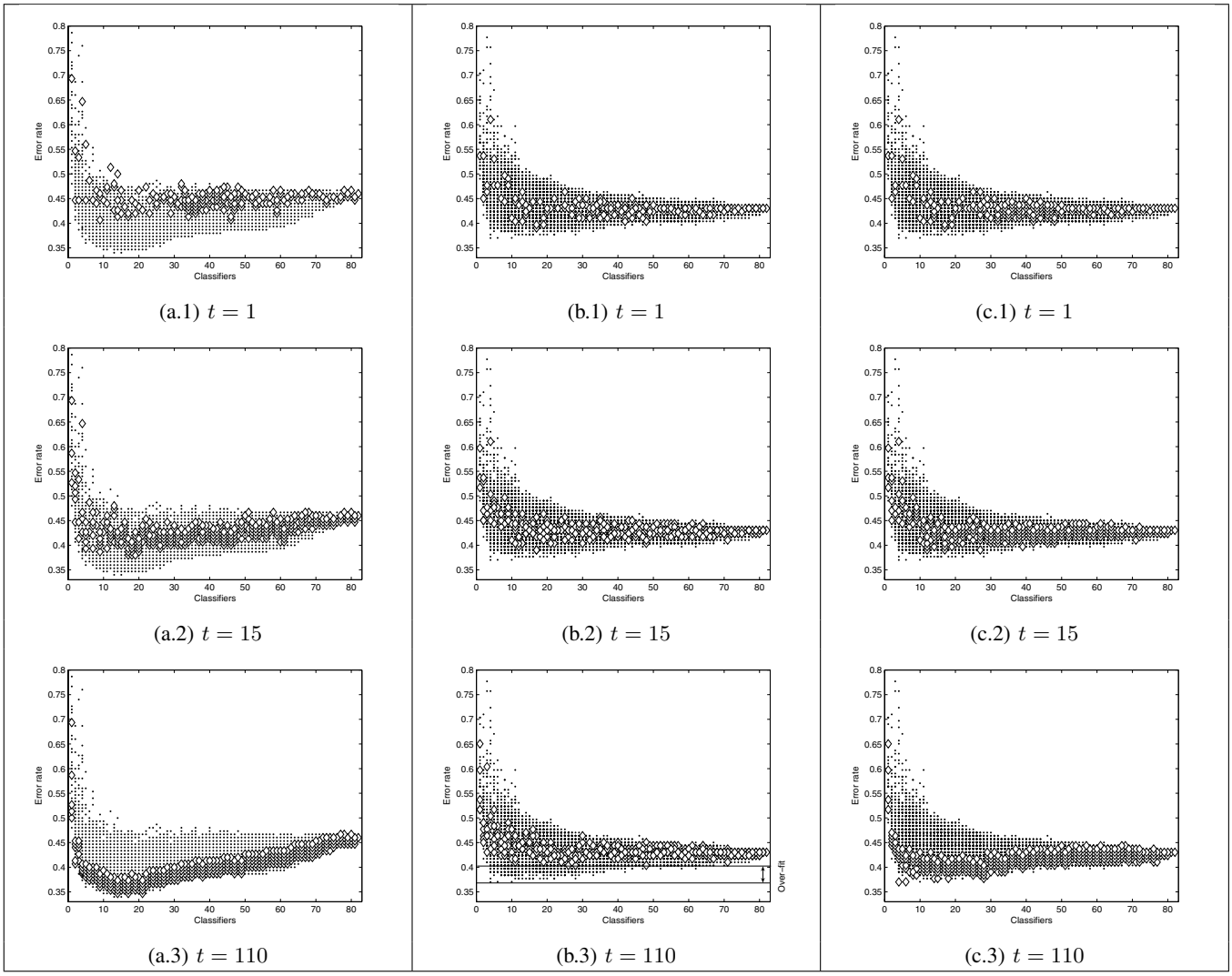


Fig. 5. EoC optimization with MOMA at generation t: (a) decision frontier on the *optimization* data set; (b) decision frontier projected on the *selection* data set; and (c) the actual decision frontier in the *selection* data set. The most accurate solution in b.3 is over-fitted by 9.75% in comparison to the most accurate solution in c.3.

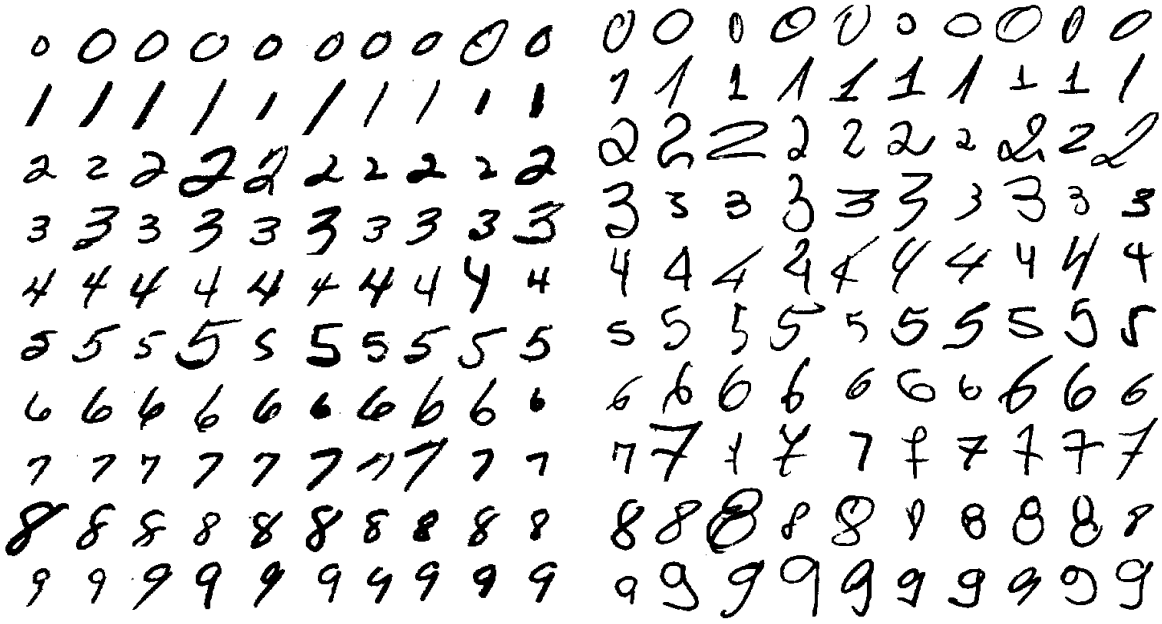
IV. Experimental Protocol and Results

To investigate the effectiveness of the over-fit control strategy, experiments are conducted for three different situations: (1) no validation stage is used and solutions are selected based only on the *optimization* data set; (2) candidate solutions are validated at the last iteration using the *selection* data set; (3) candidate solutions are validated at all iterations using the *selection* data set (global validation). The three approaches are compared in order to determine which produces the best results.

In this work, handwritten digits classification is chosen as the example application. Image-based pattern recognition (PR) requires that pixel information be first transform into an abstract representation (a feature vector) suitable for automatic recognition with classifiers [11], [12]. For isolated handwritten symbols, the choice takes into account the *domain context*, which type of symbols will be classified, and the *domain knowledge*, that is, what has been done previously to solve similar problems. Such a process is usually performed by a human expert on a trial-and-error basis. To minimize the burden on the human expert in defining and adapting classifiers, an evolutionary multi-objective optimization problem (MOOP) can be formulated. Figure 6 illustrates the difficult task of handwritten digits recognition. Both image groups (a) and (b) belong to the same PR problem but the differences in writing style require different representations for higher classification accuracy.

The goal in the experiments is to optimize a single classifier and an EoC as MOOPs. Single classifiers and EoC are both used for classification, but in different situations. A single classifier is faster and suitable for classification systems running on hardware with limited processing resources, such as embedded devices. An EoC demands more processing power and is adequate for high-performance and robust classification systems running on desktop or server computers.

The disjoint data sets in Table I are used in the experi-



(a) (b)

Fig. 6. Handwritten digits: (a) NIST SD-19, and (b) Brazilian checks.

ments. They are isolated handwritten digits extracted from NIST-SD19. In the following experiments a MLP (Multi-Layer Perceptron) neural networks classifier training is performed with the *training* data set, while a PD classifier is trained with the smaller *training'* data set.

TABLE I
HANDWRITTEN DIGITS DATA SETS EXTRACTED FROM NIST-SD19.

Data set	Size	Origin	Sample range
training'	50000	hsf_0123	1 to 50000
training	150000	hsf_0123	1 to 150000
validation	15000	hsf_0123	150001 to 165000
optimization	15000	hsf_0123	165001 to 180000
selection	15000	hsf_0123	180001 to 195000
$test_a$	60089	hsf_7	1 to 60089
$test_b$	58646	hsf_4	1 to 58646

The *validation* data set is used to adjust the classifier parameters (MLP hidden nodes and PD hyperplanes) used during optimization. The wrapper approach is performed with the *optimization* data set. And the *selection* data set is used to validate candidate solutions. Finally, solutions are compared with $test_a$ and $test_b$ data sets containing unknown observations to optimized solutions. It is known that $test_b$ is more difficult to classify than $test_a$ [13], hence the efficiency of the resulting solutions are tested on two different levels of classification complexity.

Both the NSGA-II and MOMA algorithms are used to optimize cardinality $|C|$ (features number for IFE and classifier number for EoC) and the associated classification error rate on the *optimization* data set. Their parameters are the following: population size $m = 64$ for IFE and $m = 166$ for EoC optimization, crossover probability $pc = 0.8$, mutation probability $pm = 1/L$, where L is the length of the mutated binary string [14]. MOMA specific parameter values for local search are: $n = 1$ neighbors, $NI = 3$ iterations and deviation $a = 0\%$. The maximum number of iterations is set to 1000 for all experiments with 30 replications per experiment. Computations are conducted on a Beowulf cluster with 25 nodes using Athlon XP 2500+ processors with 1 GB of RAM per node. The optimization process is implemented using LAM/MPI v6.5.

In terms of validation strategy, the results indicate an order relation between the approaches tested. Using no validation is worse than using validation at the last generation, which in turn is worse than using the proposed global validation strategy. Mean error rate values in Tables II and III are lower with the proposed global validation strategy for the PD classifier. As the original MOMA included an auxiliary archive for validation at the last generation, results to optimize a single PD (IFE problem) with this optimizer are the same with both validation at the last generation and global validation. This effect is caused by the objective function space associated to the IFE problem, which is not as large as the EoC optimization problem, where the global validation produces a lower average error rate.

The same trend is observed in Table IV and V for the MLP classifier in single and EoC configurations.

To illustrate the actual error rate of solutions obtained, error rate dispersion for all tests is detailed as box plots

TABLE II

SINGLE PD OPTIMIZATION RESULTS – MEAN VALUES ON 30 REPLICATIONS.

Validation	NSGA-II			MOMA		
	$\|C\|$	$test_a$	$test_b$	$\|C\|$	$test_a$	$test_b$
None	264	2.57%	6.42%	264	2.57%	6.42%
Last generation	220	2.44%	6.14%	330	2.18%	5.47%
Global	347	2.22%	5.55%	330	2.18%	5.47%

TABLE III

PD EOC OPTIMIZATION RESULTS – MEAN VALUES ON 30 REPLICATIONS.

Validation	NSGA-II			MOMA		
	$\|C\|$	$test_a$	$test_b$	$\|C\|$	$test_a$	$test_b$
None	12	2.08%	5.40%	17	2.07%	5.32%
Last generation	11	2.07%	5.37%	26	2.01%	5.17%
Global	25	2.00%	5.19%	22	1.98%	5.14%

TABLE IV

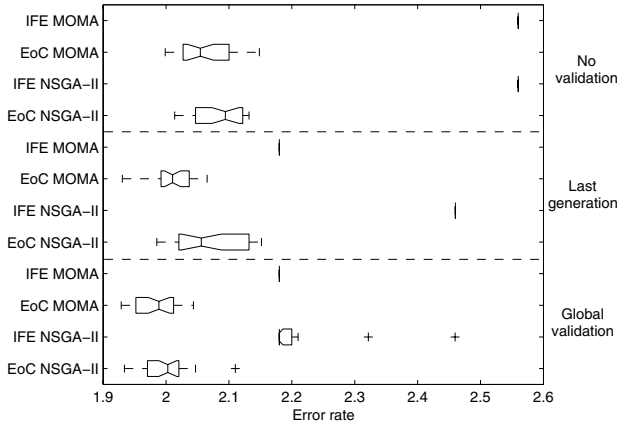
SINGLE MLP OPTIMIZATION RESULTS – MEAN VALUES ON 30 REPLICATIONS.

Validation	NSGA-II			MOMA		
	$\|C\|$	$test_a$	$test_b$	$\|C\|$	$test_a$	$test_b$
None	132	0.98%	2.81%	176	0.93%	2.84%
Last generation	264	0.91%	2.56%	330	0.82%	2.51%
Global	301	0.83%	2.52%	330	0.82%	2.51%

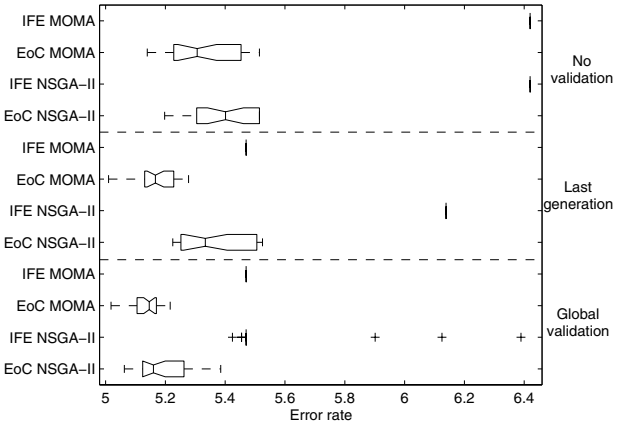
TABLE V

MLP EOC OPTIMIZATION RESULTS – MEAN VALUES ON 30 REPLICATIONS.

Validation	NSGA-II			MOMA		
	$\|C\|$	$test_a$	$test_b$	$\|C\|$	$test_a$	$test_b$
None	10	0.78%	2.44%	7	0.77%	2.41%
Last generation	5	0.77%	2.42%	16	0.76%	2.37%
Global	14	0.76%	2.36%	10	0.77%	2.35%



(a) $test_a$



(b) $test_b$

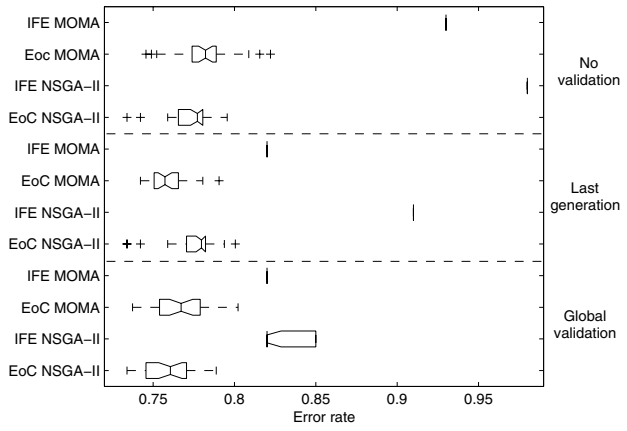
Fig. 7. PD error rate dispersion.

in Figs. 7 and 8 (PD and MLP classifiers respectively). The trend observed with average values is also observed in both plots. One excepetion is the EoC optimization with MOMA, where validation at the last generation performs similarly to global validation with both classifiers, thanks to the archive originally embedded in MOMA for validation after optimization. However, the impact of solution over-fit in not known *a priori*, and seems to depend on the problem complexity and the configuration of the optimizer employed – a smaller archive size with the original MOMA would yield different results. Thus, it is safer to use the global validation strategy in this context, as expected error rates are lower in all situations.

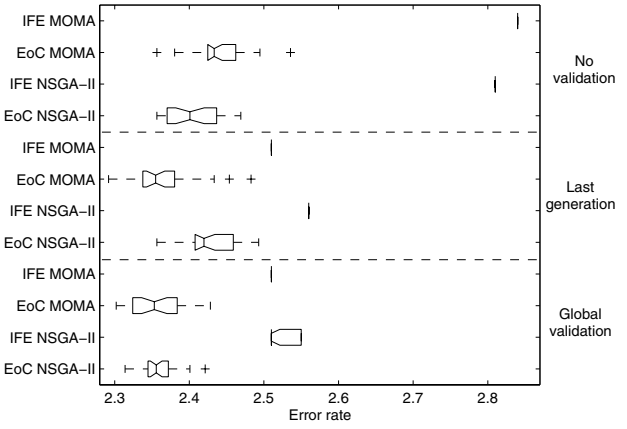
The conclusions discussed for the validation strategies were also verified in a multiple comparison. A Kruskal-Wallis nonparametric test is used to test the equality of mean values, using bootstrap to create the confidence intervals from the 30 observations in each sample. The conclusions presented regarding the validation strategies were confirmed as true, with a confidence level of 95% ($\alpha = 0.05$).

V. CONCLUSIONS

This work demonstrated that, similar to learning algorithms, methodologies to optimize classification system using a wrapped classifier are prone to solution over-fit. Validation strategies to overcome this problem have been discussed and tested. Experimental tests were used to evaluate the over-fit control strategies in optimizing classification systems with both the PD and MLP classifiers, in both single and EoC configurations. It was observed from the results that since the impact of over-fit on solutions is not known *a priori*, it is better to use a validation stage to assure solution quality during the optimization process. Given the two validation strategies to control over-fit, results indicate that the global validation is the better approach to guarantee solution quality.



(a) $test_a$



(b) $test_b$

Fig. 8. MLP error rate dispersion.

ACKNOWLEDGMENTS

The first author would like to acknowledge the CAPES and the Brazilian government for supporting this research through scholarship grant BEX 2234/03-3. The other authors would like to acknowledge the NSERC (Canada) for supporting this research.

REFERENCES

[1] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[2] P. V. W. Radtke, T. Wong, and R. Sabourin, "A Multi-Objective Memetic Algorithm for Intelligent Feature Extraction," in *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*. Berlin: Springer-Verlag, 2005, pp. 767–781.

[3] P. V. W. Radtke, R. Sabourin, and T. Wong, "Intelligent feature extraction for ensemble of classifiers," in *Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR 2005)*. IEEE Computer Society, 2005, pp. 866–870.

[4] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II," in *Proceedings of the Parallel Problem Solving from Nature VI Conference*, Paris, France, 2000, pp. 849–858.

[5] F. Kimura, S. Inoue, T. Wakabayashi, S. Tsuruoka, and Y. Miyake, "Handwritten Numeral Recognition using Autoassociative Neural Networks," in *Proceedings of the International Conference on Pattern Recognition*, 1998, pp. 152–155.

[6] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "A Methodology for Feature Selection Using Multi-Objective Genetic Algorithms for Handwritten DigitString Recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 6, pp. 903–929, 2003.

[7] C. Emmanouilidis, A. Hunter, and J. MacIntyre, "A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator," in *Proceedings of the 2000 Congress on Evolutionary Computation CEC00*. La Jolla Marriott Hotel La Jolla, California, USA: IEEE Press, 6-9 2000, pp. 309–316.

[8] G. Tremblay, R. Sabourin, and P. Maupin, "Optimizing nearest neighbour in random subspaces using a multi-objective genetic algorithm," in *17th International Conference on Pattern Recognition – ICPR2004*. Cambridge, U.K.: IEEE Computer Society, August 2004, pp. 208–211.

[9] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[10] T. G. Dietterich, "Ensemble Learning," in *The Handbook of Brain Theory and Neural Networks*, 2nd ed., M. A. Arbib, Ed. MIT Press, 2002.

[11] ϕivind Dur trier, A. K. Jain, and T. Taxt, "Feature Extraction Methods for Character recognition – A Survey," *Pattern Recognition*, vol. 29, no. 4, pp. 641–662, 1996.

[12] Z.-C. Li and C. Y. Suen, "The partition-combination method for recognition of handwritten characters," *Pattern Recognition Letters*, vol. 21, no. 8, pp. 701–720, 2000.

[13] P. J. Grother, *NIST Special Database 19 – Handprinted forms and characters database*, National Institute of Standards and Technoloy – NIST, 1995, database CD documentation.

[14] Ágoston Endre Eiben, R. Hinterdind, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124–141, 1999.