SIFT-SS: An Advanced Steady-State Multi-Objective Genetic Fuzzy System

Michel González[1], Jorge Casillas[2], and Carlos Morell[1]

[1] Universidad Central "Marta Abreu" de Las Villas, CUBA
{michelgb,cmorellp}@uclv.edu.cu
[2] Dept. Computer Science and Artificial Intelligence, University of Granada, Spain
casillas@decsai.ugr.es

Abstract. Nowadays, automatic learning of fuzzy rule-based systems is being addressed as a multi-objective optimization problem. A new research area of multi-objective genetic fuzzy systems (MOGFS) has capture the attention of the fuzzy community. Despite the good results obtained, most of existent MOGFS are based on a gross usage of the classic multi-objective algorithms. This paper takes an existent MOGFS and improves its convergence by modifying the underlying genetic algorithm. The new algorithm is tested in a set of real-world regression problems with successful results.

1 Introduction

In the last few years have grown the number of publications, in which automatic learning of fuzzy rule-based systems (FRBSs) is defined as a multi-objective optimization problem [1,2]. In the current approach, several interpretability and accuracy metrics are optimized during the learning process and higher quality models are obtained.

Multi-objective genetic fuzzy systems (MOGFSs) are playing a fundamental role in this quest. They take the best of two research fields. On the one hand, genetic fuzzy systems (GFS) have a solid base of data structures and coding schemes that can be used to simultaneously learn many features of the FRBS. On the other hand, multi-objective evolutionary algorithms (MOEAs) are among the best and more versatile techniques for multi-objective optimization.

Despite the advances obtained, there is still much work to be done. Most of the existent MOGFSs are based on a gross usage of standard MOEAs like NSGA-II [3] and SPEA2. However, there are recent results about current MOEAs limitations and new multi-objective techniques that require attention from the MOGFS's community [4]. Besides, there are specific problem requirements in fuzzy modeling that can be taken into account to enhance the search process [5].

This paper is a study case in which the specificities of fuzzy modeling are incorporated into an existent MOGFS. For the study we focus on the algorithm SIFT [6] (Section 2) due to the high number of optimization objectives, its good performance and its standard MOEA core based on NSGA-II.

E.S. Corchado Rodriguez et al. (Eds.): HAIS 2010, Part II, LNAI 6077, pp. 1–8, 2010.
© Springer-Verlag Berlin Heidelberg 2010

The proposed SIFT-SS (Section 3) introduces the following modifications in the original algorithm.

– The generational scheme is changed to an iterational scheme.
– A new Objective Scale Crowding Distance is introduced (Section 3.1).
– A new Crowding-Based Mating heuristic is introduced (Section 3.2).
– The population size is dynamically adjusted (Section 3.3).
– The phenotypical copies are removed (Section 3.4).

These modifications do not interfere with the algorithm's specific components, thus they can be easily implemented in other existent algorithms.

2 SIFT

Simplification of Fuzzy Models by Tuning, SIFT [6], is a multi-objective genetic algorithm with a generational evolutionary scheme based on NSGA-II [3]. It tunes the whole database definition (fuzzy variables, number and type of linguistic terms, and membership function parameters), while the fuzzy rule base is adapted to the tuned database by a greedy approach. The individuals are optimized according to three objectives: the mean square error (MSE) over the training dataset, the total number of linguistic terms (NL), and the number of fuzzy rules (NR). The output is a Pareto of optimal fuzzy models.

SIFT presents a set of advantages. It is a highly efficient for large-scale regressions problems. It generates very legible fuzzy partitions thanks to its interpretability constrains and the fact of tuning the complete database definition. Besides, the obtained models are small and very accurate because of the multi-objective approach.

Although SIFT is a good algorithm, its evolutionary process can be improved. As explained by Gacto [5] there are specific issues that need to be considered in the integration of MOEAs and GFS. In the case of SIFT, a careful analysis shows that two of the objectives (NL and NR) are correlated by definition and converge faster than the third objective (MSE) pulling to local minimums. This leads to many solutions with very low number of rules and poor fitness that are not very useful.

Another disadvantage of SIFT is its disability to work with large population sizes. It takes a long time for a generational algorithm like NSGA-II reach a sufficient number of iterations with a large population. Although the generational nature of SIFT allows the use of parallelism, there is not always sufficient computational resources to do it.

3 SIFT-SS: An Improved Steady-State Version of SIFT

The proposed steady-state SIFT (SIFT-SS) reuses the codification, genetic operators and evaluation of SIFT; and modifies only the underlying genetic algorithm. The changes include the substitution of the generational scheme by an iterational scheme. The iterational scheme improves convergence because each new born individual is instantly introduced in the population. As a result, it is expected a rapid advance with the same number of evaluations. The SIFT-SS core consists of the next steps:

1. Generate an initial population P and evaluate P.
2. Build a dominance rank and calculate the crowding distance for every front (see section 3.1 for objective scale crowding distance).
3. While not reached the maximun number of iterations do:
   (a) Select of two parents from P (see section 3.2 for mating heuristics).
   (b) Cross and mutate the selected parents and produce two new individuals.
   (c) Evaluate the new individuals.
   (d) Insert the new individuals in P, rebuild the rank and update crowding distance values (see section 3.4 for copies check).
   (e) Remove the worst individuals and ajust the population size (see section 3.3 for variable population size).
4. Output P

3.1 Objective Scaled Crowding Distance (OSCD)

All multi-objective evolutionary algorithms based on dominance selection sooner or later get stock when all the solutions in the population are non-dominated. The higher number of objectives, the sooner they will get stock [4].

NSGA-II is victim of this issue, in which the first selection criterion is unable to distinguish the best individuals. Of course, there is no such thing like best individuals when they are all non-dominated. The second selection criterion in NSGA-II is the Crowding Distance (CD) [3]. This is a measure which acts as an estimation of the density of solutions surrounding particular solution. When the solutions ranking is built, the most isolated ones are preferred.

The CD preserves the diversity of the population and also will lead to an equally spread solution set. But, is it always an equally spread solution set the most representative or desired? Although in other optimization problems it may be true, in the case of a rule-based system learning algorithm like SIFT the researcher may be more interested in obtaining more accurate solutions but highest number of rules rules than having very inaccurate solutions with few number of rules [5].

In this section we present a new Objective Scaled Crowding Distance (OSCD) that extends the traditional definition in order to take into account the objectives values. The OSCD will be equal to the product of the traditional CD and an Expansion Factor (K), given by the expression (1).

$$K(s) = \sum \left\{ b_s^i (e_i - 1) + 1 \right\} \tag{1}$$

where $b_s^i \in [0, 1]$ indicates the relative "goodness" of the solution s for the objective i among the rest of solutions in the same Pareto front and the parameter $e_i \in [1, \infty)$ establishes the level of strength desired for objective i (e.g. 1x, 2x, . . .).

There are many ways to measure the relative goodness of a solution for an objective. In the particular case of SIFT in which solutions with better MSEs are preferred, b_s^{MSE} is defined as the relative position of the solution in the list of solutions ordered by MSE. Therefore, the simplified expression for OSCD in SIFT is as follows:

$$OSCD(s) = \left(\frac{p_{mse}(s)}{n} (e_{mse} - 1) + 1 \right) CD(s) \tag{2}$$

with $p_{mse}(s)$ being the inverted ranking of the solution s with respect to the MSE objective (where n is the value of the most accurate solution and 1 the one of the worst).

The expansion of crowding distance allows an increased solution density in the zone with better MSE values. This allows more activity in the evolution of accuracy than in the rest of objectives, but, unlike Gacto approach [5] the OSCD is based in the control of crowding rather than $SPEA2_{Acc}$.

3.2 Crowding-Based Mating (CBM)

The Crowding-Based Mating (CBM) considers the crowding distance in order to exploit the most promising solutions. It works in the following way:

1. Select the first parent p_1 at random from the first front, using a discrete probability distribution proportional to each individual crowding distance.
2. Select the second parent p_2 by binary tournament selection.

The use of CBM in combination with OSCD guaranties that one of the two parents is likely to be an accurate and isolated solution. This press to recombine the most accurate solutions in order to obtain the least number of rules [5], but also preserves diversity. To avoid over-fitting, CBM is applied with probability P_{cbm}. Otherwise, both parents are selected by binary tournament in the same way the original SIFT does.

3.3 Variable Population Size (VPS)

The variable population size (VPS) is one of the fundamental strengths of the iterational scheme thus it can be more flexible to include a larger number of solutions without compromising too much the overall performance.

The population size in SIFT-SS can dynamically grow when all the individuals are non-dominated and dynamically shrink when dominated individuals appear. The variation is controlled between a minimum and maximum number of individuals defined by the user.

The VPS gives certain degree of flexibility to keep optimal solutions, that otherwise would have been eliminated. In problems that tends to many non-dominated solutions (either because of the large search space or the use of many objectives to be optimized) this type of population size management can be useful.

3.4 Copies Check (CC)

The SIFT-SS also implements a copy check routine that prevents the insertion of phenotypical copies[1] in the population. The decision of removing copies in SIFT-SS is sustained in two aspects. On the one hand, a copy consumes space in the population with redundant phenotypical information that does not help to the selection process. On the other hand, in the iterational scheme, copies are not as important for the survival and reproduction of elite individuals as they are in generational scheme.

4 Experimental Analysis

This section compares the original SIFT against nine different configurations of the proposed SIFT-SS. The configuration 1 was chossen to observe if the new iterational approach improves the generational approach. Configurations 2, 3 and 4 study the effect of the CC, the VPS and the OSCD heuristics separatly. Configurations 5 and 6 test the OSCD reinforced with CBM in the half and totality of crossovers respectively. Finally configurations 7, 8 and 9 are analogous to 4, 5 and 6 with the inclusion of CC. The detailed parameter specifications for SIFT-SS configurations are listed in Table 1 (if a parameter is not used then is marked with a dash).

The experimentation has been performed with a 5-fold cross validation. Each algorithm was executed with six different random seeds, so a total of 30 experiments per problem and algorithm were done. The comparison covered twelve real-world regression problems of increasing level of complexity (from 2 to 40 input variables, from 43 to 16,599 instances).

All experiments were initialized using a population of 30 individuals; the crossover probability was set to 0.7 and the mutation probability to 0.2. The stop condition was set to 50,000 evaluations. As regards the initial maximum

Table 1. Algorithm Configurations

Algorithm	P_{cbm}	E_{mse}	MCD	VPS	CC	Description
sift-ss.1	-	-	no	-	no	sift-ss
sift-ss.2	-	-	no	-	yes	sift-ss + cc
sift-ss.3	-	-	no	30-60	no	sift-ss + vps 30-60
sift-ss.4	-	2x	yes	-	no	sift-ss + oscd 2x
sift-ss.5	0.5	2x	yes	-	no	sift-ss + cbm 50% + oscd 2x
sift-ss.6	1.0	2x	yes	-	no	sift-ss + cbm 100% + oscd 2x
sift-ss.7	-	2x	yes	-	yes	sift-ss + oscd 2x + cc
sift-ss.8	0.5	2x	yes	-	yes	sift-ss + cbm 50% + oscd 2x + cc
sift-ss.9	1.0	2x	yes	-	yes	sift-ss + cbm 100% + oscd 2x + cc

[1] i.e. solutions with the same objective values.

number of linguistic terms, in the output variable it was set to seven, while in the input variable it was set to seven for problems with 2 input variables (diabetes and ele1), five for problems with 4 to 9 input variables (laser, ele2, dee, concrete, and ankara), or three for problems with 15 to 40 input variables (mortgage, treasury, elevators, compactiv, and ailerons).

To measure the convergence improvement of each algorithm we used the Generational Distance (GD) proposed by Van Veldhuizen [7] which is expressed as follows (3):

$$GD(S) = \frac{1}{|S|} \sum_{x \in S} \min \{\|f(x) - f(y)\| : y \in S^*\} \tag{3}$$

where S is the Pareto solution of the algorithm; S^* is the true Pareto-optimal solution for the problem, and $\|f(x) - f(y)\|$ is the Euclidean Distance between two solutions in the objectives space (objectives values were scale to $[0, 1]$).

Since S^* is unknown in the considered real-world problems, we use as S^* the set of non-dominated solutions among all solutions examined in our computational experiments in this paper [8], i.e., the joined Pareto obtained by all the analyzed algorithms.

The GD meassures the proximity of a solution S to the "best known" solution S^* for a problem. Considering that all algorithms produce the same number of individuals with the same genetic operators, a significant reduction in the average GD means that the proposed modification provides a better orientation in the search process.

The average values of GD for each problem are calculated in Table 2. The best result for each problem is shown underlined. At first glance, the iterational scheme by its own (configuration 1) did not show the expected improvement of GD compared to SIFT. Nevertheless, its major advantage (i.e., dealing with larger population size) was not evaluated in the experiments.

As recommended by Demšar [9], the average values of GD were statistically processed using a Friedman test. The test detected highly significant differences ($p < 0.05$) among algorithms. The mean ranks (Table 2) confirm a better performance of 8 and 9.

Next, a Wilcoxon Signed-Ranks test was applied between each pair of algorithms. Table 3 shows the summarized result. Above the diagonal; there is a "+" when the row is significant better than the column ($p < 0.05$), a "−" when the column is significant better than the row and a "=" if there is no signifcant differences between them. Below the diagonal appears the p-values of the test.

The results of Table 3 show that the full combination of OSCD, CBM and CC (8 and 9) achieve the best results. The test stands a highly significant improvement of the GD compared to the rest, except for 7.

The OSCD performs really good, whether in combination with CC or alone (4 and 7). The increased density in the zone of accurate solutions makes possible a substantial reduction of the number of rules. The effect of OSCD can be observed in Figure 1 where some of the best solutions have been plotted.

Table 2. Average Generational Distance

		SIFT-SS								
Dataset	SIFT	1	2	3	4	5	6	7	8	9
diabetes	0.0769	0.0789	0.0746	0.0895	0.0921	0.0840	0.0862	0.0783	0.0774	0.0888
ele1	0.0619	0.0557	0.0596	0.0595	0.0576	0.0609	0.0623	0.0619	0.0530	0.0589
laser	0.0906	0.0911	0.0865	0.0758	0.0791	0.0816	0.0801	0.0712	0.0767	0.0790
ele2	0.0867	0.0682	0.0770	0.0863	0.0706	0.0671	0.0691	0.0645	0.0599	0.0540
dee	0.0440	0.0393	0.0370	0.0354	0.0361	0.0364	0.0373	0.0358	0.0361	0.0343
concrete	0.0531	0.0560	0.0587	0.0587	0.0510	0.0528	0.0574	0.0535	0.0514	0.0509
ankara	0.1176	0.1314	0.1119	0.1462	0.0717	0.0912	0.0771	0.0857	0.0589	0.0613
mortgage	0.1546	0.1921	0.1379	0.1824	0.1533	0.1171	0.1387	0.1601	0.1333	0.1178
treasury	0.0922	0.1091	0.1146	0.1660	0.0914	0.0862	0.0817	0.0751	0.0896	0.0625
elevator	0.0863	0.0808	0.0522	0.0782	0.0506	0.0718	0.0747	0.0668	0.0585	0.0587
compactiv	0.0471	0.0498	0.0486	0.0505	0.0507	0.0515	0.0470	0.0473	0.0424	0.0405
ailerons	0.0653	0.0453	0.0526	0.0562	0.0481	0.0569	0.0489	0.0429	0.0436	0.0473
Mean Ranks	7.50	7.00	6.17	7.50	5.17	5.83	6.08	4.33	2.67	2.75

Table 3. Wilcoxon signed-rank test (p-values)

	8	9	7	4	5	6	2	1	S	3
SIFT-SS (8)		=	=	+	+	+	+	+	+	+
SIFT-SS (9)	0.937		=	+	+	+	+	+	+	+
SIFT-SS (7)	0.136	0.099		=	=	=	=	+	=	+
SIFT-SS (4)	0.019	0.028	0.695		=	=	=	=	+	+
SIFT-SS (5)	0.050	0.012	0.084	0.638		=	=	=	+	=
SIFT-SS (6)	0.012	0.005	0.209	0.937	1.000		=	=	+	+
SIFT-SS (2)	0.012	0.034	0.272	0.209	0.272	0.347		=	=	+
SIFT-SS (1)	0.002	0.019	0.012	0.099	0.182	0.209	0.530		=	=
SIFT (S)	0.003	0.010	0.060	0.034	0.015	0.023	0.084	1.000		=
SIFT-SS (3)	0.005	0.006	0.005	0.034	0.071	0.034	0.034	0.209	0.480	

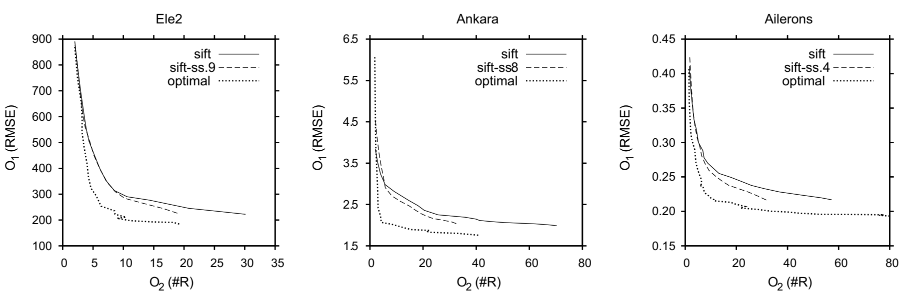

Fig. 1. Average Pareto solutions

The CBM reinforced the effect of the OSCD, but this only led to a better performance if the copies were avoided by the CC approach. In general, the removal of copies CC was found important when using OSCD or CBM as can be observed in configurations 7, 8 and 9 compared to 4, 5 and 6.

The VPS approach did not make any difference. Due to the interpretability constrains of SIFT, the number of non-dominated was not high and the modification was not effective.

5 Conclusion and Further Work

We have proposed an improved MOFGS by modifying the underlying genetic algorithm to consider the specific needs of fuzzy modeling. The proposed SIFT-SS implements heuristics like OSCD and CBM that allow a better trade-off between the accuracy and interpretability objectives. In the future this heuristics are going to be analyzed in the generational approach of SIFT. Also there is going to be further experimentation with larger search spaces and many objectives to test the true capacities of the iterational approach and the VPS.

References

1. Ishibuchi, H.: Multiobjective genetic fuzzy systems: Review and future research directions. In: Proc. of 2007 IEEE International Conference on Fuzzy Systems, London, UK, July 23-26, pp. 913–918 (2007)
2. Ishibuchi, H.: Evolutionary multiobjective design of fuzzy rule-based systems. In: Proc. of 2007 IEEE Symposium on Foundation of Computational Intelligence, Honolulu, USA, April 1-5, pp. 9–16 (2007)
3. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
4. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary Many-Objective Optimization. In: Proc. of 3rd International Workshop on Genetic and Evolving Fuzzy Systems, Witten-Bommerholz, Germany, pp. 47–52 (2008)
5. Gacto, M.J., Alcalá, R., Herrera, F.: Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems. Soft Comput. 13(5), 419–436 (2008)
6. Casillas, J.: Efficient multi-objective genetic tuning of fuzzy models for large-scale regression problems. In: Proc. of 2009 IEEE International Conference on Fuzzy Systems, Jeju, Republic of Korea, pp. 1712–1717 (2009)
7. Van Veldhuizen, D.A.: Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. Ph.D. dissertation. Air Force Institute of Technology, Dayton (1999)
8. Ishibuchi, H., Narukawa, K., Tsukamoto, N., Nojima, Y.: An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization. European Journal of Operational Research 188(1), 57–75 (2008)
9. Demšar, J.: Statistical Comparisons of Classifiers over Multiple Data Sets. Journal of Machine Learning Research 7, 30 (2006)