

Designing a Phenotypic Distance Index for Radial Basis Function Neural Networks

Jesús González, Ignacio Rojas, Héctor Pomares and Julio Ortega

Department of Computer Architecture and Computer Technology
University of Granada

Abstract. MultiObjective Evolutionary Algorithms (MOEAs) may cause a premature convergence if the selective pressure is too large, so, MOEAs usually incorporate a niche-formation procedure to distribute the population over the optimal solutions and let the population evolve until the Pareto-optimal region is completely explored. This niche-formation scheme is based on a distance index that measures the similarity between two solutions in order to decide if both may share the same niche or not. The similarity criterion is usually based on a Euclidean norm (given that the two solutions are represented with a vector), nevertheless, as this paper will explain, this kind of metric is not adequate for RBFNNs, thus being necessary a more suitable distance index. The experimental results obtained show that a MOEA including the proposed distance index is able to explore sufficiently the Pareto-optimal region and provide the user a wide variety of Pareto-optimal solutions.

1 Introduction

The automatic optimization of a RBFNNs from training data [8, 9, 18, 19] is a problem in which two clearly competing objectives must be satisfied. The model's prediction error must be minimized in order to achieve a well fitted model, while the number of Radial Basis Functions (RBFs) should be as low as possible to obtain a reliable interpolator. The problem here is how to satisfy both objectives simultaneously. Improving one of them will probably worsen the other. This kind of problem is known as a Multi-Objective Problem (MOP) [16, 11, 2, 10], and their solutions are usually sub-optimal for each objective in particular, but "acceptable" taking all the objectives into account, where "acceptable" is totally subjective and problem-dependent.

The algorithms proposed in the literature to construct RBFNNs from examples usually try to find a unique model with a compromise between its complexity and its prediction error. This is not an adequate approach. In MOPs there is usually more than one alternative optimal solution (each making different compromises between multiple objectives) that can be considered equivalent. Thus, it is very difficult to adapt conventional optimization techniques to solve MOPs because they were not designed to deal with more than one solution simultaneously [6]. Nevertheless, Evolutionary Algorithms (EAs) maintain a population of potential solutions for the problem, thus making it easier to adapt them to

solve MOPs [6]. In particular, the fitness of the solutions must be adapted to comprise all the objectives to be satisfied and new mutation operators must be designed to alter the structure of RBFNNs. With these changes, an EA becomes a MOEA able to search the Pareto-optimum region and find a wide variety of solutions with different compromises between the competing objectives.

2 Evolving competing objectives

The great difference between single objective and multiple objective problems is that the set of solutions is not completely ordered for MOPs. In the case of RBFNNs, as the complexity of a net gets higher, it can achieve a lower approximation error, but it loses generalization properties, so if we want to satisfy both objectives (low approximation error and high generalization properties), there will exist a set of Pareto-optimal solutions with different compromises between the competing objectives. Thus, good multiobjective algorithm should find as many Pareto-optimum solutions as possible, to provide the final user with the possibility of choosing the right solution following his own criteria.

There have been proposed several approaches in the literature to adapt EA to MOPs, such as the MOGA presented in [6] or the NSGA described in [21]. In [8] we used the approach proposed in [6], which defines the concept of *rank* of a solution as:

$$\text{rank}(\iota_j) = 1 + \text{dom}_j \quad (1)$$

where dom_j represents the number of solutions dominating ι_j in the current population¹. Note that rank improves when becomes smaller, that is, as the rank of ι_j gets a lower value, ι_j represents a better solution for the MOP, thus, all the Pareto-optimum solutions will be assigned a rank value of one.

This simple modification in the fitness evaluation of the RBFNNs allows a generic EA to solve an MOP transparently, that is, without changing any other of its components, although the design of expert evolutionary operators can improve the search results, as reported in [9]. The shortcoming of this scheme is that at the end of the ranking procedure there may exist many solutions having the same rank, so, the selection procedure will maintain or delete blocks of solutions for the next generation, what is likely to produce a large selection pressure which may cause a premature convergence of the algorithm [7].

To avoid this premature convergence problem, MOGA and our MOEA [8] use a niche-formation method to distribute the population over the Pareto-optimal region. The difference between MOGA and our MOEA is that the former performs a fitness sharing step based on the objective function values and our MOEA uses the parameter values for this step, thus maintaining the diversity in the parameter set rather than in the objective function values. This change allows to distinguish two different solution with the same fitness value, but needs

¹ A solution ι_j dominates another solution ι_k if ι_j is better or equal than ι_k for all the objectives in the MOP (approximation error and generalization properties)

a phenotypic distance index to estimate the similarity between two solutions of the problem.

Two solutions are considered belonging to the same niche if their phenotypic distance is lower than the threshold σ_{sh} [4]. The idea behind this sharing scheme is to divide the population into several niches formed by similar solutions, and modify the fitness of the members of a niche inversely proportional to the size of the niche. To carry out this task, the following equation is applied [21]:

$$Sh(d(\iota_j, \iota_k)) = \begin{cases} 1 - \left(\frac{d(\iota_j, \iota_k)}{\sigma_{sh}}\right)^2, & \text{if } d(\iota_j, \iota_k) < \sigma_{sh} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where $d(\iota_j, \iota_k)$ denotes the phenotypic distance index between ι_j and ι_k described below.

Finally, the fitness of every solution is modified according to:

$$f'_j = f_j \cdot \sum_{k=1}^n Sh(d(\iota_j, \iota_k)) \quad (3)$$

being n the size of the population.

3 A Phenotypic Distance Index for RBFNNs

Most of the approaches in the literature use a Euclidean norm in the parameter space to estimate the phenotypic distance between two solutions, given that these two solutions are coded as arrays of parameters [5, 4, 21], but this metric is not suitable for RBFNNs. Using this metric, two identical RBFNNs with the same RBFs (centers and radii), but coded in different positions in the array of RBFs, would obtain a distance different than 0. This simple example demonstrate the necessity of a more adequate phenotypic distance for this problem.

Another issue to consider is that in the MOP that we are solving, two different solutions may have different complexity (number of RBFs), thus, the distance index must be designed having this fact in mind. Taking these considerations into account, this paper proposes the following phenotypic distance index:

$$d(\iota_j, \iota_k) = \frac{\sum_{i_1=1}^{m_j} \min \{ \|c_{j_{i_1}} - c_{k_{i_2}}\| : 1 \leq i_2 \leq m_k \}}{m_j + m_k} + \frac{\sum_{i_2=1}^{m_k} \min \{ \|c_{k_{i_2}} - c_{j_{i_1}}\| : 1 \leq i_1 \leq m_j \}}{m_j + m_k} \quad (4)$$

where m_j and m_k are respectively the number of RBFs in ι_j and ι_k , and $c_{j_{i_1}}$ y $c_{k_{i_2}}$ their centers. Basically, this index calculates the Euclidean norm between

each center in one RBFNN and its closest center the other RBFNN, and then calculates the mean value of this measures.

The proposed phenotypic index is not affected by the coding order of the RBFs in the solutions, different number of RBFs in the nets and it is also robust against RBFNNs with several identical RBFs. Thus, it is able to distribute the solutions over the Pareto-optimal region and provide the user a wide variety of possibilities, as shown in the following section.

4 Experimental Results

As described above, the MOEA [8] with the new phenotypic distance index is able to obtain in only one execution several optimum solutions for different configurations (a Pareto-optimum frontier of solutions) for a given training set of examples. In this case, the examples belong to the Mackey-Glass time-delay differential equation [13]:

$$\frac{ds(t)}{dt} = \alpha \cdot \frac{s(t - \tau)}{1 + s^{10}(t - \tau)} - \beta s(t) \quad (5)$$

Following previous studies [22], the parameters were fixed to $\alpha = 0.2$, $\beta = 0.1$, thus obtaining a chaotic time series with no clearly defined period; it does not converge or diverge, and is very sensitive to initial conditions.

As in [12], the time series values at integer points were obtained applying the fourth-order Runge-Kutta method to find the numerical solution for the above equation. The values $s(0) = 1.2$, $\tau = 17$, and $s(t) = 0$ for $t < 0$ were assumed. This data set can be found in the file `mgdata.dat` belonging to the FUZZY LOGIC TOOLBOX OF MATLAB 5.

Following the conventional settings to perform a long term prediction of these time series, we predict the value $s(t + 85)$ from the current value $s(t)$ and the past values $s(t - 6)$, $s(t - 12)$, and $s(t - 18)$; thus, the training vectors for the model have the following format:

$$[s(t - 18), s(t - 12), s(t - 6), s(t), s(t + 85)] \quad (6)$$

As argued by Moody and Darken in [15], this prediction problem is a significant challenge in which classical methods do little better than chance, and thus the use of RBFNNs is justified.

The first 500 input vectors were used to train the model and the next 500 vectors were used to test the RBFNNs obtained. The algorithm was run several times with a population of 25 solutions for 1000 generations, and the Levenberg-Marquardt minimization algorithm was applied to the best solutions found to fine-tune their parameters. Table 1 compares the obtained result with other approaches presented in the literature in terms of their *Root Mean Squared Error* (RMSE), defined as:

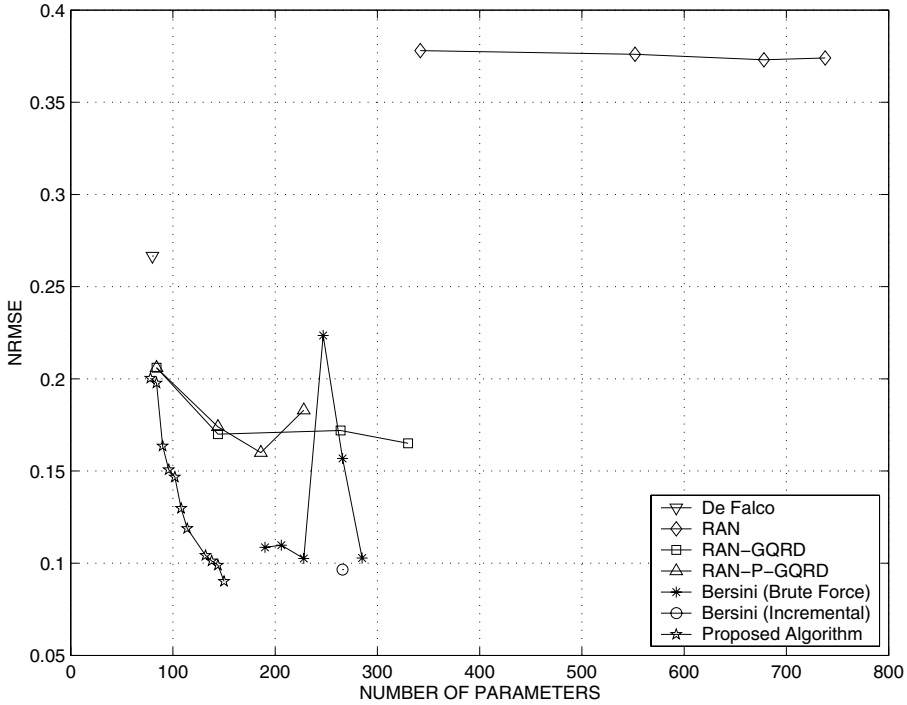


Fig. 1. Comparison of the proposed algorithm with others applied in the literature to predict the $s(t + 85)$ value of the Mackey-Glass time series

$$RMSE = \sqrt{\frac{\sum_{k=1}^n (y^k - \mathcal{F}(\mathbf{x}^k; \Phi, \Omega))^2}{n}} \tag{7}$$

Some of the other approaches are also based on RBFNNs, such as the model RAN [17], which iteratively constructs an RBFNN analyzing the novelty of the input data, or the modifications of RAN proposed in [20], which include the Givens QR decomposition (RAN-GQRD) to obtain the weights of the net and a pruning criterion (RAN-P-GQRD) to reduce the complexity of the net. The results are compared with other paradigms too. One of them [1] presents two different algorithms to train fuzzy systems, one using brute force and another incremental, and it is shown that the brute force approach presents an unstable behavior as the number of rules is increased and it not reaches the approximation errors obtained by the incremental algorithm. The other one [3] applies BGAs (*Breeder Genetic Algorithms*) to train MLPs. It can be appreciated that the algorithm with the new phenotypic distance index is able to distribute the

Algorithm		m	n_p	Test NRMSE
MLP + BGA (De Falco <i>et al.</i> 1998)		16	80	0.2666
RAN (Platt 1991)	$\epsilon = 0.1$	57	342	0.378
	$\epsilon = 0.05$	92	552	0.376
	$\epsilon = 0.02$	113	678	0.373
	$\epsilon = 0.01$	123	738	0.374
RAN-GQRD (Rosipal <i>et al.</i> 1998)	$\epsilon = 0.1$	14	84	0.206
	$\epsilon = 0.05$	24	144	0.170
	$\epsilon = 0.02$	44	264	0.172
	$\epsilon = 0.01$	55	330	0.165
RAN-P-GQRD (Rosipal <i>et al.</i> 1998)	$\epsilon = 0.1$	14	84	0.206
	$\epsilon = 0.05$	24	144	0.174
	$\epsilon = 0.02$	31	186	0.160
	$\epsilon = 0.01$	38	228	0.183
Fuzzy Systems (Bersini <i>et al.</i> 1997)	Brute Force	10	190	0.1086
		11	206	0.1098
		12	228	0.1026
		13	247	0.2235
		14	266	0.1568
		15	285	0.1028
	Incremental	14	266	0.0965
Proposed Algorithm		13	78	0.2003 ± 0.0178
		14	84	0.1977 ± 0.0164
		15	90	0.1635 ± 0.0401
		16	96	0.1507 ± 0.0193
		17	102	0.1467 ± 0.0178
		18	108	0.1297 ± 0.0175
		19	114	0.1188 ± 0.0131
		20	120	0.1268 ± 0.0174
		21	126	0.1187 ± 0.0104
		22	132	0.1042 ± 0.0135
		23	138	0.1012 ± 0.0132
24	144	0.0989 ± 0.0063		
25	150	0.0901 ± 0.0066		

Table 1. Comparison of the proposed algorithm with others applied in the literature to predict the $s(t + 85)$ value of the Mackey-Glass time series; m represents the number of RBFs or rules (depending on the model), and n_p is the number of free parameters

population over the Pareto-optimal solutions that dominate all the solutions in Table 1. Figure 1 summarizes graphically the results.

5 Conclusions

This paper presents a new phenotypic distance index to improve the fitness sharing procedure in a MOEA. The objective of this new index is to better distribute the solutions in the population over the Pareto-optimal region and thus avoid the premature convergence of the algorithm.

Section 4 has shown that with this new index, a wide variety of RBFNNs have been obtained, each one Pareto-optimal, but with a different compromise between the two conflicting objectives: approximation error and complexity.

Acknowledgment

This work has been partially supported by the Spanish *Ministerio de Ciencia y Tecnología*, under project DPI2001-3219.

References

1. H. Bersini, A. Duchateau, and N. Bradshaw. Using Incremental Learning Algorithms in the Search for Minimal and Effective Fuzzy Models. In *Proceedings of the 6th IEEE International Conference on Fuzzy Systems*, pages 1417–1422, Barcelona, Spain, July 1997. IEEE Computer Society Press.
2. V. Chankong and Y. Y. Haimes. *Multiobjective Decision Making Theory and Methodology*. North-Holland, New York, 1983.
3. I. De Falco, A. Della Cioppa, A. Iazzetta, P. Natale, and E. Tarantino. Optimizing Neural Networks for Time Series Prediction. In R. Roy, T. Furuhashi, and P. K. Chawdhry, editors, *Proceedings of the 3rd On-line World Conference on Soft Computing (WSC3). Advances in Soft Computing - Engineering Design and Manufacturing*, Internet, June 1998. Springer Verlag.
4. K. Deb. Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design. In K. Miettinen, P. Neittaanmäki, M. M. Mäkelä, and J. Périaux, editors, *Proceedings of Evolutionary Algorithms in Engineering and Computer Design, EUROGEN'99*. John Wiley, Apr. 1999.
5. K. Deb and D. E. Goldberg. An Investigation of Niche and Species Formation in Genetic Function Optimization. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 42–50, San Mateo, CA, 1989. Morgan Kaufmann.
6. C. M. Fonseca and P. J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423. Morgan Kaufmann, 1993.
7. D. E. Goldberg and K. Deb. A COMPARISON OF SELECTION SCHEMES USED IN GENETIC ALGORITHMS. pages 69–93. Morgan Kaufmann, San Mateo, CA, 1991.

8. J. González, I. Rojas, H. Pomares, and J. Ortega. RBF Neural Networks, Multiobjective Optimization and Time Series Forecasting. In Mira and Prieto [14], pages 498–505.
9. J. González, I. Rojas, H. Pomares, and M. Salmerón. Expert Mutation Operators for the Evolution of Radial Basis Function Neural Networks. In Mira and Prieto [14], pages 538–545.
10. A. E. Hans. Multicriteria optimization for highly accurate systems. In W. Stadler, editor, *Multicriteria Optimization in Engineering and Sciences, Mathematical Concepts and Methods in Science and Engineering*, volume 19, pages 309–352, New York, 1988. Plenum Press.
11. C. L. Hwang and A. S. M. Masud. *Multiple Objective Decision Making – Methods and Applications*, volume 164 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, Berlin, 1979.
12. J. S. R. Jang. ANFIS: Adaptive Network-based Fuzzy Inference System. *IEEE Trans. Syst., Man, Cybern.*, 23:665–685, May 1993.
13. M. C. Mackey and L. Glass. Oscillation and Chaos in Physiological Control Systems. *Science*, 197(4300):287–289, 1977.
14. J. Mira and A. Prieto, editors. *Connectionist Models of Neurons, Learning Processes and Artificial Intelligence*, volume 2084 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001.
15. J. E. Moody and C. J. Darken. Fast Learning in Networks of Locally-Tuned Processing Units. *Neural Computation*, 1:281–294, 1989.
16. V. Pareto. *Cours D'Economie Politique*, volume I and II. F. Rouge, Lausanne, 1896.
17. J. Platt. A Resource Allocation Network for Function Interpolation. *Neural Computation*, 3:213–225, 1991.
18. I. Rojas, J. González, A. Cañas, A. F. Díaz, F. J. Rojas, and M. Rodriguez. Short-Term Prediction of Chaotic Time Series by Using RBF Network with Regression Weights. *Int. Journal of Neural Systems*, 10(5):353–364, 2000.
19. I. Rojas, H. Pomares, J. González, J. L. Bernier, E. Ros, F. J. Pelayo, and A. Prieto. Analysis of the Functional Block Involved in the Design of Radial Basis Function Networks. *Neural Processing Letters*, 12(1):1–17, Aug. 2000.
20. R. Rosipal, M. Koska, and I. Farkaš. Prediction of Chaotic Time-Series with a Resource-Allocating RBF Network. *Neural Processing Letters*, 7:185–197, 1998.
21. N. Srinivas and K. Dev. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, 1995.
22. B. A. Whitehead and T. D. Choate. Cooperative-Competitive Genetic Evolution of Radial Basis Function Centers and Widths for Time Series Prediction. *IEEE Trans. Neural Networks*, 7(4):869–880, July 1996.