

# Scalable Continuous Multiobjective Optimization with a Neural Network–Based Estimation of Distribution Algorithm

Luis Martí, Jesús García, Antonio Berlanga, and José M. Molina

Universidad Carlos III de Madrid, Group of Applied Artificial Intelligence  
Av. de la Universidad Carlos III, 22 — Colmenarejo, Madrid 28270, Spain  
{lmarti,jgherrer}@inf.uc3m.es, {aberlan,molina}@ia.uc3m.es  
<http://www.giaa.inf.uc3m.es/>

**Abstract.** To achieve a substantial improvement of MOEDAs regarding MOEAs it is necessary to adapt their model building algorithm to suit this particular task. Most current model building schemes used so far off-the-shelf machine learning methods. However, the model building problem has specific requirements that those methods do not meet and even avoid.

In this we work propose a novel approach to model building in MOEDAs using an algorithm custom-made for the task. We base our proposal on the growing neural gas (GNG) network. The resulting model-building GNG (MB-GNG) is capable of yielding good results when confronted to high-dimensional problems.

## 1 Introduction

Although multiobjective optimization evolutionary algorithms (MOEAs) [1] have successfully solved many complex synthetic and real-life problems, the majority of works have been concentrated on low dimensional problems [2]. When advancing to higher dimensions MOEAs suffer heavily from the curse of dimensionality [3], requiring an exponential increase of the resources made available to them (see [1] pp. 414–419).

A possible way of alleviating this scaling issue is to resort to more efficient evolutionary approaches. Estimation of distribution algorithms (EDAs) [4] can be used for that objective. EDAs have been hailed as landmark in the progress of evolutionary algorithms. They replace the application of evolutionary operators with the creation of a statistical model of the fittest elements of the population. This model is then sampled to produce new elements.

The extension of EDAs to the multiobjective domain has lead to what can be denominated multiobjective EDAs (MOEDAs). However most MOEDAs have limited themselves to port single objective EDAs to the multiobjective domain by incorporating features taken from MOEAs. Although MOEDAs have proved themselves as a valid approach to the MOP, this later fact hinders the achievement of a significant improvement regarding “standard” MOEAs in high-dimensional problems.

Adapting the model building algorithm is one of the ways of achieving a substantial advance. Most model building schemes used so far by EDAs use off-the-shelf machine learning methods. However, the model building problem has specific requirements that those methods do not meet and even avoid. In particular, in those algorithms, outliers are treated as invalid data, where in the model building problem they represent newly discovered regions of the search space. Similarly, an excess of resources is spent in finding the optimal size of the model.

In this work we propose a novel approach to model building in MOEDAs using an algorithm custom-made for the task. We modify the growing neural gas (GNG) network [5] to make it particularly suitable for model building. The resulting model-building GNG (MB-GNG) addresses the above described issues with success. Therefore it is capable of yielding better results when confronted to high-dimensional problems.

The rest of this work first deals with the theoretical background that supports our proposal. We then proceed to describe MB-GNG and the EDA framework in which it is embedded. After that, a series of well-known test problems are solved with a set of other state-of-the-art algorithms and our approach. Each problem is scaled regarding the number of optimization functions in order to assess the behavior of the algorithms when subjected to extreme situations. As a conclusion some final remarks, comments and lines of future development are presented.

## 2 Theoretical Background

The concept of multiobjective optimization refers to the process of finding one or more feasible solutions of a problem that corresponds to the extreme values (either maximum or minimum) of two or more functions subject to a set of restrictions. More formally, a multiobjective optimization problem (MOP) can be defined as

**Definition 1 (Multiobjective Optimization Problem).**

$$\left. \begin{array}{l} \text{minimize } \mathbf{F}(\mathbf{x}) = \langle f_1(\mathbf{x}), \dots, f_M(\mathbf{x}) \rangle, \\ \text{subject to } c_1(\mathbf{x}), \dots, c_R(\mathbf{x}) \leq 0, \\ \text{with } \mathbf{x} \in \mathcal{D}, \end{array} \right\} \quad (1)$$

where  $\mathcal{D}$  is known as the *decision space*. The functions  $f_1(\mathbf{x}), \dots, f_M(\mathbf{x})$  are the objective functions. Their corresponding image set,  $\mathcal{O}$ , of  $\mathcal{D}$  is named *objective space* ( $\mathbf{F} : \mathcal{D} \rightarrow \mathcal{O}$ ). Finally, inequalities  $c_1(\mathbf{x}), \dots, c_R(\mathbf{x}) \leq 0$  express the restrictions imposed to the values of  $\mathbf{x}$ .

In general terms, this class of problems does not have a unique optimal solution. Instead an algorithm solving (1) should produce a set containing equally good trade-off optimal solutions. The optimality of a set of solutions can be defined relying on the so-called *Pareto dominance relation* [6]:

**Definition 2 (Dominance Relation).** *Under optimization problem (1) and having  $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D}$ ,  $\mathbf{x}_1$  is said to dominate  $\mathbf{x}_2$  (expressed as  $\mathbf{x}_1 \prec \mathbf{x}_2$ ) iff  $\forall f_j, f_j(\mathbf{x}_1) \leq f_j(\mathbf{x}_2)$  and  $\exists f_i$  such that  $f_i(\mathbf{x}_1) < f_i(\mathbf{x}_2)$ .*

The solution of (1) is a subset of  $\mathcal{D}$  that contains elements are not dominated by other elements of  $\mathcal{D}$ , or, in more formal terms,

**Definition 3 (Pareto-optimal Set).** *The solution of problem (1) is the set  $\mathcal{D}^*$  such that  $\mathcal{D}^* \subseteq \mathcal{D}$  and  $\forall \mathbf{x}_1 \in \mathcal{D}^* \nexists \mathbf{x}_2$  that  $\mathbf{x}_2 \prec \mathbf{x}_1$ .*

Here,  $\mathcal{D}^*$  is known as the *Pareto-optimal set* and its image in objective space is called *Pareto-optimal front*,  $\mathcal{O}^*$ .

## 2.1 Evolutionary Approaches to Multiobjective Optimization

MOPs have been addressed with a variety of methods [7]. Among them, evolutionary algorithms (EAs) [8] have proved themselves as a valid and competent approach from theoretical and practical points of view. This has led to what has been called multiobjective optimization evolutionary algorithms (MOEAs) [1]. Their success is due to the fact that EAs do not make any assumptions about the underlying fitness landscape. Therefore, it is believed they perform consistently well across all types of problems, although it has been shown that they share theoretical limits imposed by the no-free-lunch theorem [9]. Another important benefit arises from the parallel search. Thanks to that these algorithms can produce a set of equally optimal solutions instead of one, as many other algorithms do.

## 2.2 Estimation of Distribution Algorithms

Estimation of distribution algorithms (EDAs) have been claimed as a paradigm shift in the field of evolutionary computation. Like EAs, EDAs are population based optimization algorithms. However in EDAs the step where the evolutionary operators are applied to the population is substituted by construction of a statistical model of the most promising subset of the population. This model is then sampled to produce new individuals that are merged with the original population following a given substitution policy. Because of this model building feature EDAs have also been called probabilistic model building genetic algorithms (PMBGAs) [10]. A framework similar to EDAs is proposed by the iterated density estimation evolutionary algorithms (IDEAs) [11].

The introduction of machine learning techniques implies that these new algorithms lose the biological plausibility of its predecessors. In spite of this, they gain the capacity of scalably solve many challenging problems, significantly outperforming standard EAs and other optimization techniques.

First EDAs were intended for combinatorial optimization but they have been extended to continuous domain (see [4] for a review). It was then only a matter of time to have multiobjective extensions. This has led to the formulation of multiobjective optimization EDAs (MOEDAs).

### 3 Scalability: A Salient Issue

One topic that remains not properly studied inside the MOEA/MOEDA scope is the scalability of the algorithms [12]. The most critical issue is the dimension of the objective space. It has been experimentally shown to have an exponential relation with the optimal size of the population (see [1] pp. 414–419). This fact implies that, with the increase of the number of objective functions an optimization algorithm needs an exponential amount of resources made available to it.

In order to achieve a substantial improvement in the scalability of these algorithms it is therefore essential to arm them with a custom-made model building algorithm that pushes the population towards newly found zones in objective space. So far, MOEDA approaches have mostly used off-the-shelf machine learning methods. However, those methods are not meant specifically for the problem we are dealing with here. This fact leads to some undesirable behaviors that, although justified in the original field of application of the algorithms, might hinder the performance of the process, both in the accuracy and in the resource consumption senses. Among these behaviors we can find the creation of an excessively accurate model and the loosing of outliers.

For the model building problem there is no need for having the most accurate model of the data. However, most of the current approaches dedicate a sizable effort in finding the optimal model complexity. For instance, for cluster-based models its not required to find optimal amount of clusters just to find a “fair” amount such that the data set to be modeled is correctly covered.

In the other case, outliers are essential in model building. They represent unexplored areas of the local optimal front. Therefore they not only should be preserved but perhaps even reinforced. A model building algorithms that primes outliers might actually accelerate the search process and alleviate the rate of the exponential dimension–population size dependency.

In the next section we introduce a MOEDA that uses a custom-made model building algorithm to overcome the problems here described.

### 4 Model Building Growing Neural Gas

Clustering algorithms [13] have been used as part of the model building algorithms of EDAs and MOEDAs. However, as we discussed in the previous section a custom-made algorithm might be one of the ways of achieving a significative improvement in this field.

For this task we have chosen the growing neural gas (GNG) network [5] as a starting point. GNG networks are intrinsic self-organizing neural networks based on the neural gas [14] model. This model relies in a competitive Hebbian learning rule [15]. It creates an ordered topology of inputs classes and associates a cumulative error to each. The topology and the cumulative errors are conjointly used to determine how new classes should be inserted. Using these heuristics the model can fit the network dimension to the complexity of the problem being solved.

Our model building GNG (MB-GNG) is an extension of the original (unsupervised) GNG. MB-GNG is a one layer network that defines each class as a local Gaussian density and adapts them using a local learning rule. The layer contains a set of nodes  $\mathcal{C} = \{c_1, \dots, c_{N^*}\}$ , with  $N_0 \leq N^* \leq N_{\max}$ . Here  $N_0$  and  $N_{\max}$  represent initial and maximal amount of nodes in the network.

A node  $c_i$  consists of a center,  $\boldsymbol{\mu}_i$ , deviations,  $\boldsymbol{\sigma}_i$ , an accumulated error,  $\xi_i$ , and a set of edges that define the set of topological neighbors of  $c_i$ ,  $\mathcal{V}_i$ . Each edge has an associated age,  $\nu_{i,j}$ .

MB-GNG creates a quantization of the inputs space using a modified version of the GNG algorithm and then computes the deviations associated to each node. The dynamics of a GNG network consists of three concurrent processes: network adaptation, node insertion and node deletion. The combined use of these three processes renders GNG training Hebbian in spirit [15].

The network is initialized with  $N_0$  nodes with their centers set to randomly chosen inputs. A training iteration starts after an input  $\mathbf{x}$  is randomly selected from the training data set. Then two nodes are selected for being the closest ones to  $\mathbf{x}$ . The *best-matching node*,  $c_b$ , is the closest node to  $\mathbf{x}$ . Consequently, the *second best-matching node*,  $c_{b'}$ , is determined as the second closest node to  $\mathbf{x}$ .

If  $c_{b'}$  is not a neighbor of  $c_b$  then a new edge is established between them  $\mathcal{V}_b = \mathcal{V}_b \cup \{c_{b'}\}$  with zero age,  $\nu_{b,b'} = 0$ . If, on the other case,  $c_{b'} \in \mathcal{V}_b$  the age of the corresponding edge is reset  $\nu_{b,b'} = 0$ .

At this point, the age of all edges is incremented in one. If an edge is older than the maximum age,  $\nu_{i,j} > \nu_{\max}$ , then the edge is removed. If a node becomes isolated from the rest it is also deleted.

A clustering error is then added to the best-matching node error accumulator,  $\xi_b$ .

After that, learning takes place in the best-matching node and its neighbors with rates  $\epsilon_{\text{best}}$  and  $\epsilon_{\text{vic}}$ , respectively. For  $c_b$  adaptation follows the rule originally used by GNG,

$$\Delta \boldsymbol{\mu}_b = \epsilon_{\text{best}} (\mathbf{x} - \boldsymbol{\mu}_b). \quad (2)$$

However for  $c_b$ 's neighbors a cluster repulsion term [16] is added to the original formulation, that is,  $\forall c_v \in \mathcal{V}_b$ ,

$$\Delta \boldsymbol{\mu}_v = \epsilon_{\text{vic}} (\mathbf{x} - \boldsymbol{\mu}_v) + \beta \exp \left( -\frac{\|\boldsymbol{\mu}_v - \boldsymbol{\mu}_b\|}{\zeta} \right) \frac{\sum_{c_u \in \mathcal{V}_b} \|\boldsymbol{\mu}_u - \boldsymbol{\mu}_b\|}{|\mathcal{V}_b|} \frac{(\boldsymbol{\mu}_v - \boldsymbol{\mu}_b)}{\|\boldsymbol{\mu}_v - \boldsymbol{\mu}_b\|}, \quad (3)$$

Here  $\beta$  is an integral multiplier that defines the amplitude of the repulsive force while  $\zeta$  controls the weakening rate of the repulsive force with respect to the distance between the nodes' centers. We have set them to  $\beta = 2$  and  $\zeta = 0.1$  as suggested in [17].

After a given amount of iterations a new node is inserted to the network. First, the node with largest error,  $c_e$ , is selected the node. Then the worst node among its neighbors,  $c_{e'}$ , is located.  $N^*$  is incremented and the new node,  $c_{N^*}$ , is inserted equally distant from the two nodes. The edge between  $c_e$  and  $c_{e'}$  is

removed and two new edges connecting  $c_{N^*}$  with  $c_e$  and  $c_{e'}$  are created. Finally, the errors of all nodes are decreased by a factor  $\delta_G$ ,

$$\xi_i = \delta_G \xi_i, \quad i = 1, \dots, N^*. \quad (4)$$

After training has ended the deviations,  $\sigma_i$ , of the nodes must be computed. For this task we employ the unbiased estimator of the deviations.

## 5 Embedding MB-GNG in an EDA Framework

To test MB-GNG is it essential to insert it in an EDA framework. This framework should be simple enough to be easily understandable but should also have a sufficient problem solving capacity. It should be scalable and preserve the diversity of the population.

Our EDA employs the fitness assignment used by the NSGA-II algorithm [18] and constructs the population model by applying MB-GNG. The NSGA-II fitness assignment was chosen because of its proven effectivity and its relative low computational cost.

The algorithm's workflow is similar to other EDAs. It maintains a population of individuals,  $P_t$ , where  $t$  is the current iteration. It starts from a random initial population  $P_0$  of  $z$  individuals. It then proceeds to sort the individuals using the NSGA-II fitness assignment function. A set  $\hat{P}_t$  containing the best  $\lfloor \alpha |P_t| \rfloor$  elements is extracted from the sorted version of  $P_t$ ,

$$|\hat{P}_t| = \alpha |P_t|. \quad (5)$$

A MB-GNG network is then trained using  $\hat{P}_t$  as training data set. In order to have a controlled relation between size of  $\hat{P}_t$  and the maximum size of the network,  $N_{\max}$ , these two sizes are bound,

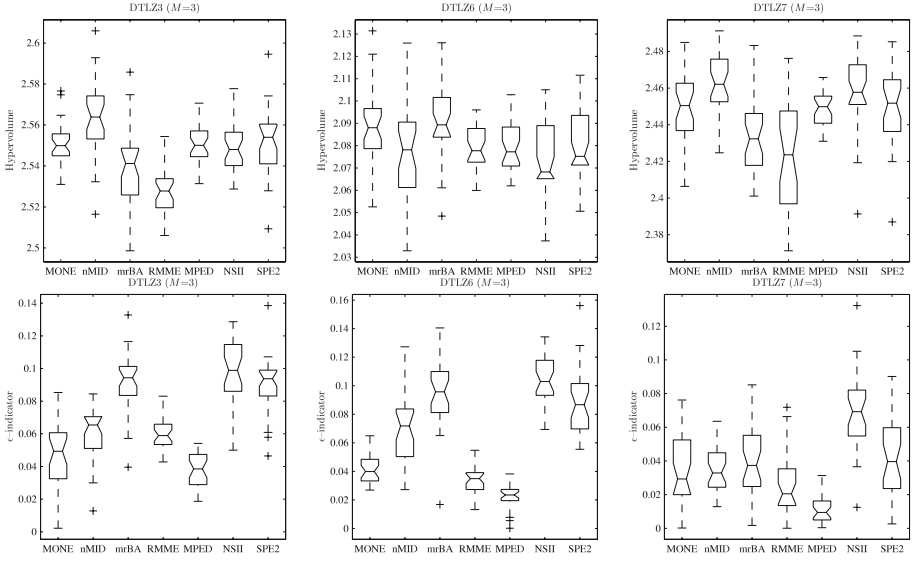
$$N_{\max} = \lfloor \gamma |\hat{P}_t| \rfloor. \quad (6)$$

The resulting Gaussian kernels are used to sample new individuals. An amount of  $\lfloor \omega |P_t| \rfloor$  new individuals is synthesized. Each one of these individuals substitute a randomly selected ones from the section of the population not used for model building  $P_t \setminus \hat{P}_t$ . The set obtained is then united with best elements,  $\hat{P}_t$ , to form the population of the next iteration  $P_t$ .

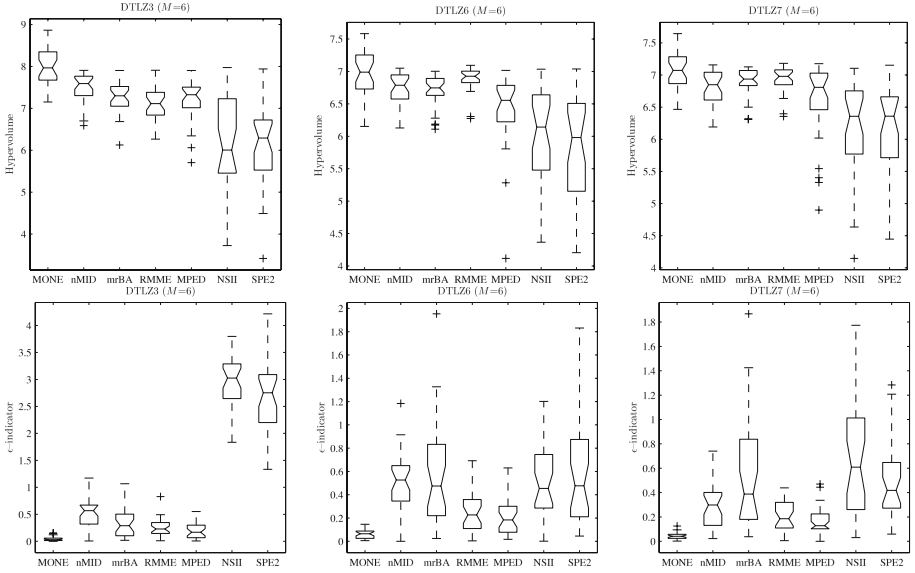
Iterations are repeated until a given stopping criterion is met. The output of the algorithm is the set of non-dominated individuals of  $P_t$ .

## 6 Experiments

We now focus on making a preliminary study that validates our proposal from an experimental point of view. For comparison purposes a set of known and



(a) Three-objective problems.



(b) Six-objective problems.

**Fig. 1.** DTLZ3, DTLZ6 and DTLZ7 test problems with naïve MIDEA (nMID), mrBOA (mrBA), RM-MEDA (RMME), MOPED (MPED), NSGA-II (NSII), SPEA2 (SPE2) and our approach (MONE)

competent evolutionary multiobjective optimizers are also applied. In particular, naïve MIDEA [19], mrBOA [20], RM-MEDA [21], MOPED [22], NSGA-II, SPEA2 [23].

As we already commented, most experiments involving MOPs deal with two or three objectives problems [2]. Instead we intended to address higher-dimensional problems, as we are interested in assessing the scalable behavior of our algorithm. Consequently we have chosen for our analysis some of the DTLZ family of scalable multiobjective problems, in particular DTLZ3, DTLZ6 and DTLZ7 [24]. Each problem was configured with 3 and 6 objective functions. The dimension of the decision space was set to 10.

Tests were carried out under the PISA experimental framework [25]. Algorithm' implementations were adapted from the ones provided by their respective authors with the exception of NSGA-II and SPEA2, that are already distributed as part of the framework; MOPED, that was implemented from scratch, and, of course, our approach.

The hypervolume and the unary additive epsilon indicators [26] were used to assess the performance. These indicators measure how close the set of solutions is to the Pareto-optimal front. The hypervolume indicator should be maximized and the unary epsilon indicator should be minimized.

Figure 1 shows the box plots obtained after 30 runs of each algorithm for solving the different the problem/dimension configuration.

In the three dimensional problems our approach performed similarly to the rest of the algorithms. This was an expected outcome since our MOEDA uses an already existent fitness function and its model building algorithm is meant to provide a significant advantage in more extreme situations.

That is the case of the six objective problem. Here our algorithm outperforms the rest of the optimizers applied. Disregarding the problem and metric chosen our approach yields better results. One can hypothesize that, in this problem, the model building algorithm induces the exploration of the search space and therefore it manages to discover as much as possible of the Pareto-optimal front. It is most interesting that our proposal exhibits rather small standard deviations. This means that it performed consistently well across the different runs. These results must be investigated further to understand if the low dispersion of the error indicators can only be obtained in the problems solved or if can be extrapolated to other problems.

## 7 Conclusions

In this work we have introduced MB-GNG, a custom-made growing neural gas neural network for model-building. As it has been particularly devised for the EDA model building problem, it is capable of yielding better results than similar algorithms. A set of well known test problems were solved. The results helped to assert MB-GNG advantages but further tests are obviously necessary. The study here presented must be extended to higher dimensions and other test problems must be solved. However, because of length restrictions we have had to limit ourselves to providing a brief outline of our current research.

These tests will cast light upon the actual processes take place during the model building phase. They should also lead to further improvements on the



model building algorithms applied. Regarding this, some other machine learning methods must be extrapolated to this domain in order to compare and establish the validity of our idea. It is expected that these new approaches will have the search capabilities of MB-GNG while minimizing its computational footprint.

## Acknowledgements

This work was supported in part by projects MADRINET, TEC2005-07186-C03-02, SINPROB, TSI2005-07344-C02-02 and CAM CCG06-UC3M/TIC-0781. The authors wish to thank the anonymous reviewers for their insightful comments.

## References

1. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester (2001)
2. Deb, K., Saxena, D.K.: On finding Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. Technical Report 2005011, KanGAL (December 2005)
3. Bellman, R.E.: Adaptive Control Processes. Princeton University Press, Princeton (1961)
4. Larrañaga, P., Lozano, J.A. (eds.): Estimation of Distribution Algorithms. A new tool for Evolutionary Computation. Genetic Algorithms and Evolutionary Computation. Kluwer Academic Publishers, Boston/Dordrecht/London (2002)
5. Fritzsche, B.: A growing neural gas network learns topologies. In: Tesauro, G., Touretzky, D.S., Leen, T.K. (eds.) Advances in Neural Information Processing Systems, vol. 7, pp. 625–632. MIT Press, Cambridge (1995)
6. Pareto, V.: Cours D'Economie Politique. F. Rouge, Lausanne (1896)
7. Miettinen, K.: Nonlinear Multiobjective Optimization. International Series in Operations Research & Management Science, vol. 12. Kluwer, Norwell (1999)
8. Bäck, T.: Evolutionary algorithms in theory and practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford University Press, New York (1996)
9. Corne, D.W., Knowles, J.D.: No Free Lunch and Free Leftovers Theorems for Multiobjective Optimisation Problems. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 327–341. Springer, Heidelberg (2003)
10. Pelikan, M., Goldberg, D.E., Lobo, F.: A survey of optimization by building and using probabilistic models. IlliGAL Report No. 99018, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL (1999)
11. Bosman, P.A.: Design and Application of Iterated Density-Estimation Evolutionary Algorithms. PhD thesis, Universiteit Utrecht, Utrecht, The Netherlands (2003)
12. Coello Coello, C.A.: Evolutionary multiobjective optimization: A historical view of the field. IEEE Computational Intelligence Magazine 1(1), 28–36 (2006)
13. Xu, R., Wunsch II, D.: Survey of clustering algorithms. IEEE Transactions on Neural Networks 16(3), 645–678 (2005)
14. Martinetz, T.M., Berkovich, S.G., Shulten, K.J.: Neural-gas network for vector quantization and its application to time-series prediction. IEEE Transactions on Neural Networks 4, 558–560 (1993)

15. Martinetz, T.M.: Competitive Hebbian learning rule forms perfectly topology preserving maps. In: International Conference on Artificial Neural Networks (ICANN 1993), Amsterdam, pp. 427–434. Springer, Heidelberg (1993)
16. Timm, H., Borgelt, C., Doring, C., Kruse, R.: An extension to possibilistic fuzzy cluster analysis. *Fuzzy Sets and Systems* 147(1), 3–16 (2004)
17. Qin, A.K., Suganthan, P.N.: Robust growing neural gas algorithm with application in cluster analysis. *Neural Networks* 17(8–9), 1135–1148 (2004)
18. 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)
19. Bosman, P.A., Thierens, D.: The Naïve MIDEA: A baseline multi-objective EA. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 428–442. Springer, Heidelberg (2005)
20. Ahn, C.W.: *Advances in Evolutionary Algorithms. Theory, Design and Practice*. Springer, Heidelberg (2006)
21. Zhang, Q., Zhou, A., Jin, Y.: A regularity model based multiobjective estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation* (in press 2007)
22. Costa, M., Minisci, E., Pasero, E.: An hybrid neural/genetic approach to continuous multi-objective optimization problems. In: Apolloni, B., Marinaro, M., Tagliaferri, R. (eds.) WIRN 2003. LNCS, vol. 2859, pp. 61–69. Springer, Heidelberg (2003)
23. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In: Giannakoglou, K., Tsahalis, D., Periaux, J., Papailou, P., Fogarty, T. (eds.) EUROGEN 2001. *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, Athens, Greece, pp. 95–100 (2002)
24. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Test Problems for Evolutionary Multiobjective Optimization. *Advanced Information and Knowledge Processing*. In: *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, pp. 105–145. Springer, Heidelberg (2004)
25. Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA—A Platform and Programming Language Independent Interface for Search Algorithms. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 494–508. Springer, Heidelberg (2003)
26. Knowles, J., Thiele, L., Zitzler, E.: A tutorial on the performance assessment of stochastic multiobjective optimizers. TIK Report 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich (2006)