

Multi-objective optimization with an adaptive resonance theory-based estimation of distribution algorithm

Luis Martí · Jesús García ·
Antonio Berlanga · José M. Molina

Published online: 7 August 2012
© Springer Science+Business Media B.V. 2012

Abstract The introduction of learning to the search mechanisms of optimization algorithms has been nominated as one of the viable approaches when dealing with complex optimization problems, in particular with multi-objective ones. One of the forms of carrying out this hybridization process is by using multi-objective optimization estimation of distribution algorithms (MOEDAs). However, it has been pointed out that current MOEDAs have an intrinsic shortcoming in their model-building algorithms that hamper their performance. In this work, we put forward the argument that error-based learning, the class of learning most commonly used in MOEDAs is responsible for current MOEDA underachievement. We present adaptive resonance theory (ART) as a suitable learning paradigm alternative and present a novel algorithm called multi-objective ART-based EDA (MARTEDA) that uses a Gaussian ART neural network for model-building and a hypervolume-based selector as described for the HypE algorithm. In order to assert the improvement obtained by combining two cutting-edge approaches to optimization an extensive set of experiments are carried out. These experiments also test the scalability of MARTEDA as the number of objective functions increases.

Keywords Multi-objective optimization · Estimation of distribution algorithms · Adaptive resonance theory

Mathematics Subject Classifications (2010) 65K10 · 68T20 · 68T05

L. Martí (✉) · J. García · A. Berlanga · J. M. Molina
Group of Applied Artificial Intelligence, Universidad Carlos III de Madrid,
Av. de la Universidad Carlos III, 22. Colmenarejo, Madrid 28270, Spain
e-mail: lmarti@inf.uc3m.es

J. García
e-mail: jgherrer@inf.uc3m.es

A. Berlanga
e-mail: aberlan@ia.uc3m.es

J. M. Molina
e-mail: molina@ia.uc3m.es

1 Introduction

Multi-objective optimization problems (MOPs) are problems where a set of (conflicting) features or *objectives* must be simultaneously optimized. The solution to these problems is a set of trade-off points that represent different compromises between those objectives. Having this set of trade-offs a decision maker applies higher-order criteria in order to determine which of them should be realized in practice.

MOP related research has seen a great deal of development as a result of the involvement of the evolutionary computation community. This fact has led to the creation of *multi-objective evolutionary algorithms* (MOEAs) (cf. [20]).

There is a class of MOPs that are particularly appealing because of their inherent complexity: the so-called *many-objective problems* [44]. These are problems with a relatively large number of objectives (usually four or more). Although somewhat counterintuitive and hard to visualize for a human decision maker, these problems are not uncommon in real-life engineering practice. For example, [48] details some relevant real problems of this type.

It has been shown that “established” approaches fail to yield adequate solutions because, as more objective functions are added, the optimization algorithms suffer heavily under the curse of dimensionality [6]; requiring an exponential increase of the resources made available to them (see [30, 43, 44] and [22, pp. 414–419]).

Approaches to the many-objective issue can be grouped in three main fronts:

1. the design of better selection (fitness assignment) functions;
2. the use of objective reduction strategies, and;
3. application of better variation (search) methods.

There has been a relatively large body of work on the first two issues. For example, it has been shown that the use performance indicators, which were originally meant for contrasting results and experiment analysis, for individuals selection allows the resulting algorithm to cope better with higher dimension problems (cf. [4, 5, 49]). Similarly, some works have focused on the reduction of the amount of objectives to a minimum by eliminating redundant or irrelevant objectives (cf. [16, 17, 23]).

The third direction, however, remains relatively unexplored. Here, a viable approach is to employ cutting-edge evolutionary algorithms that could effectively deal with high-dimensional problems more efficiently.

The incorporation of learning as part of the search processes has been nominated as a viable way of achieving progress in this direction [21]. A form of carrying out this task is to resort to *estimation of distribution algorithms* (EDAs) [33]. EDAs are capable of learning the problem structure. They replace the application of evolutionary operators with the creation of a statistical model of the fittest elements of the population in a process known as *model building*. This model is then sampled to produce new elements.

In spite of the *a priori* expectations, the so-called *multi-objective EDAs* (MOEDAs) [41] have not yielded the anticipated results. This fact necessarily prompts a set of reflections regarding the causes of this underperformance.

Most MOEDAs have limited themselves to transforming single-objective EDAs into a multi-objective formulation by including an existing multi-objective fitness assignment function. It can be hypothesized that the straightforward extrapolation

might had lead to skip the fact that most current EDAs have some characteristics that hinders their capacity of handling some of the requirements of multi-objective optimization. In a previous work [35], we identified three of those characteristics, in particular, those derived from the incorrect treatment of outstanding but isolated elements of the population (outliers); the loss of population diversity, and that too much computational effort is being spent on finding an optimal population model.

The performance issue of current MOEDAs has been traced back to the their underlying learning paradigm: the *dataset-wise error minimization learning*, or *error-based learning*, for short [35]. This class of learning, in different forms, is shared by most machine learning algorithms. It implies that a model is tuned in order to minimize a global error measured across the dataset. In this type of learning isolated data is not taken into account because of their little contribution to the overall error and therefore they do not take an active part of learning process. This assertion is in part supported by the fact that most the approaches that had a better performance in comparative experiments like [35] do not exactly conform to the error-based scheme. That is why, other learning paradigms should be assessed.

Some approaches have been proposed with success with the purpose of overcoming this issue. That is the case of the model-building growing neural gas (MB-GNG) model-building algorithm [36]. However, in spite of the positive results obtained so far a more systematic approach is needed. It is important to understand the role of the learning paradigm in model building and, particularly, what are the consequences of not using error-based learning.

Adaptive resonance theory (ART) [28] is a theory of human cognition that has seen a realization as a family of neural networks. It relies on a learning scheme denominated *match-based learning* and on intrinsic topology self-organization. These features make it interesting case study as model-building approach. Match-based learning equally weights isolated and clustered data [45], and, therefore, the algorithm does not disregard outliers. Similarly, self-organization makes possible the on-the-fly determination the model complexity required to correctly represent the data set, thus eliminating the need of an external algorithm for that task.

In this work we engage and develop the argument that error-based learning, the class of learning most commonly used in MOEDAs is responsible for current MOEDA underachievement. We discuss in detail ART-based learning as a viable alternative and present a novel algorithm called *multi-objective ART-based EDA* (MARTEDA) that uses a modification of the Gaussian ART neural network [52] for model building. This novel model-building approach is complemented by the use of the hypervolume performance indicator-based selection as described for the hypervolume estimation algorithm for multi-objective optimization (HypE) [5]. This selection method, as previously commented, has been shown to yield relevant results when dealing with many-objective problems. We experimentally show that thanks to MARTEDA's novel model-building approach and an indicator-based population ranking the algorithm it is able to outperform similar MOEDAs and MOEAs.

The main contributions of this paper can be summarized as follows:

- introduction and discussion of the use of ART and its match-based learning paradigm in the EDA context;
- description of MARTEDA, a novel MOEDA based on ART, and;
- realization of an extensive experimental study that contrasts the results of MARTEDA with the current state-of-the-art MOEAs and MOEDAs.

The remaining part of the work proceeds by briefly introducing the theoretical notions required for the presentation of the proposal. After that, we examine the model-building issue. Following that, in Section 4, we discuss the nature of ART and describe the modified Gaussian ART network that is used as start point for our model-building algorithm. Subsequently, MARTEDA is introduced in Section 5, describing how the HypE selection and Gaussian ART are blended together in a MOEDA framework. Section 6 presents and discusses the results of the comparative experiments involving MARTEDA and a selection of other current state-of-the-art algorithms dealing with a set of community accepted problems. Some of these problems are configured with an progressive number of objectives (3, 6, 9, 12 and 15) in order to assess the performance of our proposal in the context of many-objective optimization. Finally, some conclusive remarks and future lines of research are outlined.

2 Theoretical foundations

Many real-world optimization problems involve more than one goal to be optimized. This type of problems is known as *multi-objective optimization problems* (MOPs). A MOP can be expressed as the problem in which a set of *objective functions* $f_1(x), \dots, f_M(x)$ should be jointly optimized;

$$\min F(x) = \langle f_1(x), \dots, f_M(x) \rangle; x \in \mathcal{S}; \quad (1)$$

where $\mathcal{S} \subseteq \mathbb{R}^n$ is known as the *feasible set* and could be expressed as a set of restrictions over the decision set, in our case, \mathbb{R}^n . The image set of \mathcal{S} produced by function vector $F(\cdot)$, $\mathcal{O} \subseteq \mathbb{R}^M$, is called *feasible objective set* or *criterion set*.

The solution to this type of problem is a set of trade-off points. The optimality of a solution can be expressed in terms of the Pareto dominance relation.

Definition 1 (Pareto dominance relation) For the optimization problem specified in (1) and having $x, y \in \mathcal{S}$, x is said to dominate y (expressed as $x \prec y$) iff $\forall f_j, f_j(x) \leq f_j(y)$ and $\exists f_i$ such that $f_i(x) < f_i(y)$.

Definition 2 (Non-dominated subset) In problem (1) and having the set $\mathcal{A} \subseteq \mathcal{S}$, $\hat{\mathcal{A}}$, the *non-dominated subset* of \mathcal{A} , is defined as

$$\hat{\mathcal{A}} = \{x \in \mathcal{A} \mid \nexists x' \in \mathcal{A} : x' \prec x\}.$$

The solution of (1) is $\hat{\mathcal{S}}$, the non-dominated subset of \mathcal{S} . $\hat{\mathcal{S}}$ is known as the *efficient set* or *Pareto-optimal set* [14]. The *Pareto-optimal front*, $\hat{\mathcal{O}}$, is the image of $\hat{\mathcal{S}}$ in the feasible objective set.

If problem (1) has certain characteristics, e. g., linearity or convexity of the objective functions or convexity of \mathcal{S} , the efficient set can be determined by mathematical programming approaches [14]. However, in the general case, finding the solution of (1) is an NP-complete problem [2]. In this case, heuristic or metaheuristic methods, like evolutionary algorithms, can be applied in order to have solutions of practical value at an admissible computational cost.

2.1 Evolutionary algorithms

In rough terms, an evolutionary algorithm (EA) is a population-based metaheuristic that can be characterized by how it implements a set of processes, in particular,

- *Mating selection*: that establishes a partial order of individuals in the population using their fitness function value as reference and determines the degree at which individuals in the population will take part in the generation of new (offspring) individuals.
- *Offspring generation (variation)*: which applies a range of evolutionary operators, like crossover, mutation, etc., to synthesize offspring individuals from the current (parent) population. This process is supposed to prime the fittest individuals so they play a bigger role in the generation of the offspring.
- *Parents and offspring combination (environmental selection)*: that merges the parent and offspring individuals to produce the population that will be used in the next iteration. This process often involves the deletion of some individuals using a given criterion in order to keep the amount of individuals below a certain threshold.

In single objective EAs there is only one objective function to be minimized or maximized. Thanks to that, it is straightforward to use that function for determining the fitness of individuals. However, in the multi-objective case, finding this scalar indicator is a complex matter since, as in any dimensionality reduction, relevant information may be lost. Furthermore, a MOP solver is not only expected to yield solutions as close as possible to the Pareto-optimal front. Its solutions should also be as diverse as possible, therefore offering a good coverage of $\hat{\mathcal{O}}$.

Because of these reasons the ranking of individuals is one of the key issues in the MOEAs' field of research. The strategies that have been proposed to circumvent this problem can be grouped in three classes:

- *Objective function aggregation*: where objective values are combined using a weighted aggregation function of either linear or non-linear nature.
- *Pareto-based ranking*: that generate an ordering of the population individuals relying on the Pareto dominance relation.
- *Indicator-based ranking*: which use the performance indicators that were originally meant for assessing MOP optimizer's performance.

Indicator-based selection seems to have a superior performance in complex and many-objective problems [50]. Among these “good performing” algorithms we can find the *hypervolume estimation algorithm for multi-objective optimization* (HypE) [5]. HypE relies on the hypervolume performance indicator for mating and environmental selection.

The *hypervolume indicator*, $I_{\text{hyp}}(\mathcal{A})$, [31, 32, 59, 60] computes the volume of the region, H , delimited by a given set of points, \mathcal{A} , and a set of reference points, \mathcal{N} .

$$I_{\text{hyp}}(\mathcal{A}) = \text{volume} \left(\bigcup_{\forall a \in \mathcal{A}; \forall n \in \mathcal{N}} \text{hypercube}(a, n) \right). \quad (2)$$

Therefore, larger values of the indicator will correspond to better solutions. It has many attractive features that had favored its application and popularity. In particular, it is the only indicator that has the properties of a metric and the only to be strictly Pareto monotonic [25, 62]. Because of these properties this indicator has been used not only for performance assessment but also as part of some evolutionary algorithms.

To measure the absolute performance of an algorithm the reference points should ideally be *nadir points*. These points are the worst elements of \mathcal{O} , or, in other words, the elements of \mathcal{O} that do not dominate any other element. To contrast the relative performance of two sets of solutions, though, one can be used as the reference set. These matters are further detailed in [31, 61].

A lot of research has focused on improving the computational complexity of this indicator [7, 9, 26, 51]. The exact computation of the algorithm has been shown to be **#P**-hard [15] in the number of objectives. **#P** problems are the analogous of **NP** for counting problems [39]. Therefore, all algorithms calculating a hypervolume must have an exponential run-time with regard to the number of objectives if **P** \neq **NP**, something that seems to be true [24].

HypE addresses the computational cost of the hypervolume indicator by proposing Monte Carlo sampling method that approximates the value of the hypervolume at a lower computational cost. This approximation, even when it might not be adequately accurate for performance assessment is usable for the fitness assignment purpose. In this work we use HypE's hypervolume-based mating selection.

2.2 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 variation 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. Therefore, a benefit of EDAs is that not only do they return a solution to a problem, but a model representing the solutions is presented as well.

Because of this model-building feature EDAs have also been called *probabilistic-model-building genetic algorithms* (PMBGAs) [40]. A framework similar to EDAs is proposed by the *iterated density estimation evolutionary algorithms* (IDEAs) [10]. Early approaches assumed that the different features of the decision variables were independent. Subsequent methods started to deal with interactions among the decision variables, first in pair-wise fashion and later in a generalized manner, using n -ary dependencies.

Multi-objective EDAs (MOEDAs) [41] are the extensions of EDAs to the multi-objective domain. Most MOEDAs consist of a modification of existing EDAs whose fitness assignment function is substituted by one taken from an existing MOEA.

There are two complementing EDA approaches for storing or representing the search individuals. One keeps a population for search individuals and, in every iteration model the most promising subset of such population and create new individuals. On the contrary, there are other approaches that store search information as the learned model. Therefore, this model is sampled and updated based on the adequacy

of the sample. The first approach is, to the best of our knowledge, the most popular one in the multi-objective context.

Methods based on regularity, like RM-MEDA and MMEA exploit the properties (those required to apply the KKT condition) of the problem being solved. In real life practice the properties it is often impossible to check if the problem meets the requirements to apply the KKT condition. Furthermore, as they model the Pareto front as an $(m-1)$ -dimensional manifold they have high computational demands as the number of objectives grows and these many-objective problems are the ones in which we are more interested.

The MMEA paper deals with a particular class of MOP, in which the dimensionalities of the Pareto front and the Pareto set manifolds are different. In this case the approximation of the Pareto front might not yield a correct approximation to the Pareto set. MMEA is and EDA that approximates the Pareto set and the Pareto front simultaneously for an MOP of the class above-described. In this regard, our approach offers a comparable solution, as we focus on the modeling and approximation of the Pareto set, and therefore, the Pareto front.

It must be pointed out, however, that EDAs are not the only form way of incorporating learning in the optimization process. There are some approaches that perform this task by providing hybrid evolutionary/machine learning method, like, for example, the *learnable evolution model* (LEM) [38]. This approach is similar in spirit to EDAs although it also incorporates some EA features. Something that complicates its application in real-world practice. Furthermore, these efforts seem to have been concentrated on single-objective optimization (c.f. [46, 47]).

3 The model-building issue

Notwithstanding the diverse efforts dedicated to providing usable model-building methods for EDAs the nature of the problem itself has received relatively low attention and, instead, most works have just used off-the-shelf machine learning algorithms. An analysis of the results yielded by current multi-objective EDAs and their scalability with regard to the number of objective leads the identification of certain issues that might be hampering the obtention of substantially better results with regard to other evolutionary approaches.

Data outliers issue is a good example of insufficient comprehension of the nature of the model-building problem. In machine-learning practice, outliers are handled as noisy, inconsistent or irrelevant data. Therefore, outlying data is expected to have little influence on the model or just to be disregarded.

However, that behavior is not adequate for model-building. In this case, is known beforehand that all elements in the data set should be take into account as they represent newly discovered or candidate regions of the search space and therefore must be explored. Therefore, these instances should be at least equally represented by the model and perhaps even reinforced.

As model-building strategies varies from EDA to EDA, it is hard to back the previous statement with a general theoretical support. In order to do so, we must define an individual z_i as the pair representing values in decision and objective sets,

$$z_i = \langle x_i, F(x_i) \rangle . \quad (3)$$

In a simplified case, we can state that model building is an unsupervised machine learning problem with learning dataset,

$$\Psi = \{x_i\}; \forall z_i = \langle x_i, F(x_i) \rangle \in \hat{\mathcal{P}}_t, \quad (4)$$

where $\hat{\mathcal{P}}_t$ is the model-building dataset which is a subset of the algorithm population at iteration t .

The machine learning algorithm tunes the model $\mathcal{M}(x, \theta, \phi)$ by adjusting its topology θ and parameters ϕ . In error-based learning this process involves the calculation of a set-wise error to which each element-wise error contribute to a different degree,

$$E_{\text{tot}} = \sum_{x_i \in \Psi} E(\mathcal{M}(x_i, \theta, \phi)) . \quad (5)$$

There are many different forms of the set-wise and element-wise errors, E_{tot} and $E(\cdot)$ respectively, but they can be formulated in a more or less similar fashion as above.

If E_{tot} is to be minimized, then θ and ϕ will be set in such way that the aggregation of element-wise contributions is as minimal as possible. As outliers, by their own definition, are rare and infrequent, their element-wise contribution to E_{tot} could be left to be relatively large as it is more convenient to focus on those that by being more popular, have a larger contribution to E_{tot} .

Therefore, model $\mathcal{M}(x, \theta, \phi)$ would end up representing more accurately elements more densely grouped than those isolated. However, as we already mentioned, in the model-building case, all elements of Ψ are important, and, perhaps, the isolated ones might be even more important than the clustered ones, as they represent locally optimal zones of the objective set that have not been properly explored.

Another weakness of most MOEDAs (and most EDAs, for that matter) is the loss of population diversity. Diversity loss can be attributed to two main causes:

- biased selection processes, and;
- incorrect model building.

As described in the previous section, the matting selection in EDAs extracts the best subset of the population to build the model. The continuous selection of the best part of the population could lead to a premature homogenisation of the population and, therefore, to the stagnation of the search process. In the second case, the loss of diversity can be traced back to the above-described outliers issue of model-building algorithms and also to the incorrect estimation or sampling of the model. This fact leads us back to the statement referring that model building has not been correctly acknowledged as a different problem with particular requirements.

A number of works that have tried to “patch” current methods and, therefore make them more suitable for this context. For example, in [54] it is proposed a method for avoiding that the variances of a multivariate Gaussian model to drop to “quickly” drop to zero. Similarly, [13] introduced a permutation sampling that eliminates the sampling errors of UMDA. Other approaches [42, 56] have tried strategies to re-inject “fresh” individuals that are kept on a evolutionary algorithm that is run in parallel.

In previous approaches to this issue [35], different alternatives to model building were compared. The outcome of these experiments supported the hypothesis the

approaches that had a better performance do not follow the error-based scheme, like the *k*-means algorithm, randomized leader algorithm and the growing neural gas network. That is why, perhaps another classes of learning, like instance-based learning or match-based learning would yield a sizable advantage.

4 Model building with Adaptive Resonance Theory

Adaptive Resonance Theory (ART) is a theory of human cognition. ART was put forward with the purpose of dealing with *stability-plasticity dilemma* [18]. This dilemma is concerned with how can a learning system be designed to remain adaptive (or plastic in ART terminology) in response to significant events and yet remain stable in response to irrelevant ones? In other words, how to switch between stable and its plastic modes to achieve stability without rigidity and plasticity without chaos? Furthermore, how can it preserve its previously learned knowledge while continuing to incorporate new information, while preventing this new learning from “overwriting” the codes previously stored?

The cornerstone property of ART systems is a pattern matching process that compares an external input with a pattern stored in the internal memory. As matching takes place, it leads either to a resonant or stable state, which persists long enough to permit learning, or to a parallel memory search. If the search ends with a positive match with stored code, the memory representation may either remain the same or incorporate new information from matched portions of the current input. If the search indicates that a new code is needed, then memory representation learns the current input. This match-based learning process is the foundation of ART code stability.

Match-based learning allows memories to change only when input from the external world is close enough to internal expectations, or when something completely new occurs. This feature makes ART systems well suited to problems that require on-line learning of large and evolving databases. This class of learning is complementary to error-based learning, which responds to a mismatch by changing memories so as to reduce the difference between a target output and an actual output, rather than by searching for a better match. Error-based learning, as explained in the previous section, minimizes is naturally suited to problems such as adaptive control and the learning of sensory-motor maps, which require ongoing adaptation to present statistics. It has been pointed out that ART networks are not suitable for this class of classical machine-learning applications [45], however, what is an inconvenience in that area is a positive feature in the case of model-building.

ART-based neural networks are capable of fast, stable, on-line, unsupervised or supervised, incremental learning, classification, and prediction following a match-based learning scheme [28]. During training, ART networks adjust previously-learned categories in response to familiar inputs, and create new categories dynamically in response to inputs different enough from those previously seen. A vigilance test allows to regulate the maximum tolerable difference between any two input patterns in a same category. As ART networks provide a compromise between adding new knowledge and improving the representation of the existing one, it could be hypothesized that, thanks to that, they would deal in a proper manner with data

outliers. When using ART, outliers would not be disregarded since the matching process could take care of determining that they are different from currently stored elements and they will not be disregarded in later learning iterations as the model creation process proceeds. This reasoning motivated our interest in developing a MOEDA that uses an ART-inspired method for model building.

4.1 Gaussian ART for model building

There are many variations of ART networks. Among them, the Gaussian ART [52] is most suitable for model building since it is capable of handling continuous data. The result of applying Gaussian ART is a set of nodes each representing a local Gaussian density. These nodes can be combined as a Gaussian mixture that can be used to synthesize new individuals.

Gaussian ART creates classes of similar inputs (Fig. 1). It has a layer of afferent or input nodes, F1, and a classification layer, F2 (we have kept this nomenclature for the sake of homogeneity with other ART networks). The F2 layer stores classes of inputs. Its activation is a combined measure of the similarity of the input and the prototype of each class, and the size of the given class.

For the model-building task we have modified the original formulation of the network to make it more suited for the task. When an input $x \in \mathbb{R}^n$ is presented to the input layer it is propagated to the F2 layer. F2 has N^* nodes, c_1, \dots, c_{N^*} with N of them committed. Each committed node models a local density of the input space

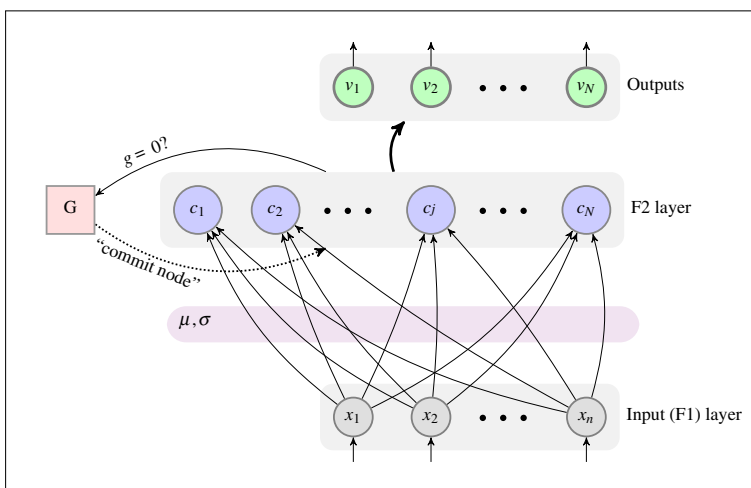


Fig. 1 Gaussian ART neural network used for model building. When an input x is presented to the input layer it is propagated to the F2 layer. Each F2 node c_j represents a local Gaussian density characterized by its weights μ_j and σ_j . The match function G_j is calculated for each node. If its value is smaller than vigilance ρ then its corresponding input strength, g_j is set to zero. If all g_j are set to zero, it means that no node could be activated and, therefore, an uncommitted node must be committed. The network output v is computed by normalizing the g_j 's. Concurrent to this learning takes place

using Gaussian receptive fields with mean μ_j and standard deviation σ_j . A node is activated if it satisfies the match criterion. That is, the match function,

$$G_j = \exp \left(-\frac{1}{2} \sum_{i=1}^n \left(\frac{x_i - \mu_{ji}}{\sigma_{ji}} \right)^2 \right), \quad j = 1, \dots, N, \quad (6)$$

must be greater than the F2 vigilance parameter, ρ ; according to this, the input strength of a node is computed as

$$g_j = \begin{cases} \frac{\eta_j}{\prod_{i=1}^n \sigma_{ji}} G_j, & \text{if } G_j > \rho \\ 0 & \text{otherwise} \end{cases}, \quad \rho > 0, \quad (7)$$

where η_j is a measure of the node a priori activation probability. This is different from the original Gaussian ART network where only one node was allowed be active after an input presentation.

After the presentation of an input, if no F2 node is active, then an uncommitted node must be committed. The task of detecting when an input is not sufficiently coded in F2 is accomplished by the F2 gain control, G , that fires if no committed nodes are active. The signal

$$\Gamma = \begin{cases} 1 & \text{if } \max_{j=1, \dots, N} g_j = 0 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

is used to commit an uncommitted node. It can also be used to offer an “I don’t know” answer during the non-adaptive use of a network.

The activation of each node is then calculated normalizing the node’s input strength,

$$v_j = \frac{g_j}{\sum_{l=1}^N g_l}. \quad (9)$$

As other ART networks, this model is an on-line learning neural network. Therefore, all adaptation processes have local rules. In F2, μ_j and σ_j are updated using a learning rule based on the gated steepest descent learning rule [27].

The gated steepest descent rule for an adaptive weight w_{ji} can be expressed as

$$\varepsilon \frac{\partial w_{ji}}{\partial t} = y_j^* [f(x_i) + w_{ji}]. \quad (10)$$

Using a neural network notation, this rule can be explained as how the post-synaptic activity, y_j^* , modulates the rate at which w_{ji} tracks the pre-synaptic signal $f(x_i)$.

The previous continuous formulation must be transformed in order to be able to apply it in an iterative fashion. The discrete-time formulation of (10) becomes

$$w_{ji}(t+1) = (1 - \varepsilon^{-1} y_j^*) w_{ji}(t) + \varepsilon^{-1} y_j^* f(x_i). \quad (11)$$

Modifying (11) we can obtain the learning equations for the F2 nodes. The constant change rate is replaced by η_j . η_j is updated to represent the cumulative category activation,

$$\eta_j(t+1) = \eta_j(t) + v_j, \quad (12)$$

and, therefore, the amount of training that has taken place in the j th node. The use of η_j equally weights inputs over time with the intention to measure their sample statistics.

The pre-synaptic signal $f(x_i)$ is substituted by x_i and x_i^2 , respectively, for learning the first and second moments of the input,

$$\mu_{ji}(t+1) = \left(1 - \eta_j^{-1} v_j\right) \mu_{ji}(t) + \eta_j^{-1} v_j x_i, \quad (13)$$

$$\lambda_{ji}(t+1) = \left(1 - \eta_j^{-1} v_j\right) \lambda_{ji}(t) + \eta_j^{-1} v_j x_i^2. \quad (14)$$

The standard deviation,

$$\sigma_{ji}(t+1) = \sqrt{\lambda_{ji}(t+1) - \mu_{ji}(t+1)^2}, \quad (15)$$

is calculated using (13) and (14) as in [53].

Gaussian ART is initialized with all nodes uncommitted ($N = 0$). Learning takes place in active ($v_j > 0$) F2 nodes following (13)–(15). However if no F2 nodes becomes active an uncommitted node is committed and therefore N is incremented. The new node is indexed by N and initialized with $v_N = 1$, $\eta_N = 0$. Learning will proceed as usual but a constant φ_i^2 will be added to each λ_{Ni} to set $\sigma_{Ni} = \varphi_i$. The value of φ_i has a direct impact on the quality of learning. A larger φ_i slows down learning in its corresponding input feature but warranties a more robust convergence.

The local Gaussian densities resulting from the described algorithm can be combined to synthesize a Gaussian mixture. This Gaussian mixture is then used can be used by the EDA to generate new individuals.

5 Multi-objective ART-based EDA

The multi-objective ART-based EDA (MARTEDA) is a MOEDA that uses the previously described Gaussian ART network as its model-building algorithm. Although it intends to deal with the issues raised by the previous discussion, it was also designed with scalability in mind, since it is expected to cope with many-objective problems. It also exhibits an elitist behavior, as it has proved itself a very advantageous property. Finally, thanks to the combination of fitness assignment and model-building it promotes diversity preservation.

MARTEDA maintains a population, \mathcal{P}_t , of n_{pop} individuals; where t is a given iteration. The algorithm's workflow is similar to other EDAs. It starts with a random initial population \mathcal{P}_0 of individuals.

At a given iteration t the algorithm determines the set $\hat{\mathcal{P}}_t$ containing the best $\lfloor \alpha |\mathcal{P}_t| \rfloor$ elements.

$$|\hat{\mathcal{P}}_t| = \lfloor \alpha |\mathcal{P}_t| \rfloor = \lfloor \alpha n_{\text{pop}} \rfloor. \quad (16)$$

$\hat{\mathcal{P}}_t$ is constructed by determining the elements of \mathcal{P}_t that produce the larger value of the hypervolume indicator, as in the HypE algorithm. For problems of two and three objectives this task is carried out by exactly calculating it. For cases of more

objectives the Monte Carlo sampling alternative is used, as it is more computational cost-effective.

A Gaussian ART network is then trained using $\hat{\mathcal{P}}_t$ as its training data set. In order to have a controlled relation between size of $\hat{\mathcal{P}}_t$ and the maximum size of the network, N_{\max} , these two sizes are bound by the rate $\gamma \in (0, 1]$,

$$N_{\max} = \left\lceil \gamma \left\lceil \hat{\mathcal{P}}_t \right\rceil \right\rceil = \left\lceil \gamma \left\lfloor \alpha n_{\text{pop}} \right\rfloor \right\rceil. \quad (17)$$

The trained network is a model of $\hat{\mathcal{P}}_t$. The network can be interpreted as a Gaussian mixture, as explained in the previous section. Therefore it can be used to sample new individuals. In particular, $\left\lfloor \omega \left\lceil \hat{\mathcal{P}}_t \right\rceil \right\rfloor$ new individuals are synthesized.

The local Gaussian densities resulting from the described algorithm can be combined to synthesize the Gaussian mixture with parameters Θ ,

$$P(x|\Theta) = \frac{1}{N} \sum_{i=1}^N P(x|\mu_i, \sigma_i). \quad (18)$$

Each Gaussian density is formulated as

$$P(x|\mu_i, \sigma_i) = \frac{1}{(2\pi)^{n/2} |\Sigma_i|^{1/2}} \exp \left(-\frac{1}{2} (x - \mu_i)^\top \Sigma_i^{-1} (x - \mu_i) \right), \quad (19)$$

with the covariance matrices Σ_i defined as a diagonal matrix with its non-zero elements set to the values of the deviations σ_i . The Gaussian mixture can be used by the EDA to generate new individuals. These new individuals are created by sampling the $P(x|\Theta)$. The generation of randomly distributed numbers that follow a given distribution has been dealt in depth by many authors. In our case, we applied the Box-Muller transformation [12].

Each one of these individuals substitute a randomly selected ones from the section of the population not used for model-building $\mathcal{P}_t \setminus \hat{\mathcal{P}}_t$. The set obtained is then united with the best elements, $\hat{\mathcal{P}}_t$, to form the population of the next iteration \mathcal{P}_{t+1} . Some other substitution strategies could be used in this step. For example, the new

Fig. 2 Algorithmic representation of MARTEDA

Parameters:

- ▷ φ , initial deviations.
- ▷ ρ , F2 vigilance threshold.
- ▷ n_{pop} , population size.
- ▷ $\alpha \in (0, 1]$, selection percentile.
- ▷ $\omega \in (0, 1]$, substitution percentile.

Algorithm:

$t \leftarrow 0$.

Randomly generate the initial population \mathcal{P}_0 with n_{pop} individuals.

repeat

Sort population \mathcal{P}_t using the HypE+ ranking algorithm.

Extract first $\alpha \left\lceil \mathcal{P}_t \right\rceil$ elements the sorted \mathcal{P}_t to $\hat{\mathcal{P}}_t$.

A Gaussian ART with $\hat{\mathcal{P}}_t$ as training data set and φ and ρ as parameters.

Sample $\left\lfloor \omega \left\lceil \mathcal{P}_t \right\rceil \right\rfloor$ from the network.

Substitute randomly selected individuals of $\mathcal{P}_t \setminus \hat{\mathcal{P}}_t$ with the new individuals to produce \mathcal{P}'_t .

$\mathcal{P}_{t+1} = \hat{\mathcal{P}}_t \cup \mathcal{P}'_t$.

$t = t + 1$.

until end condition = true

Determine the set of non-dominated individuals of \mathcal{P}_t , \mathcal{P}_t^* .

return \mathcal{P}_t^* as the algorithm's solution.

individuals could substitute the worst individuals of $\mathcal{P}_t \setminus \hat{\mathcal{P}}_t$. We have chosen the previously described approach because it promotes diversity and avoids stagnation.

Iterations are repeated until a given stopping criterion is met. The output of the algorithm is a subset of \mathcal{P}_t that contains the non-dominated solutions, \mathcal{P}_t^* (Fig. 2).

6 Experimental study

The results of the experiments involving MARTEDA, some current state-of-the-art MOEDAs and MOEAs in a selection of current community-accepted problems are reported in this section. In particular we deal with problems drawn from two current and complex problem sets: the Walking Fish Group (WFG) problems and the test instances of the CEC'09 multi-objective optimization special session and competition.

6.1 WFG problems

The Walking Fish Group (WFG) problem toolkit [29] is a toolkit for creating complex synthetic multi-objective test problems. The WFG test suite exceeds the functionality of previous existing test suites. These include: non-separable problems, deceptive problems, a truly degenerate problem, a mixed shape Pareto front problem, problems scalable in the number of position related parameters, and problems with dependencies between position- and distance-related parameters. The WFG test suite provides a better form of assessing the performance of optimization algorithms on a wide range of different problems.

From the set of nine problems WFG4 to WFG9 were selected because of the simplicity of their Pareto-optimal front that lies on the first orthant of a unit hypersphere. This decision was also caused by the high computational cost of the experiments being carried out and by the length restriction imposed upon this contribution. Each of these problem was devised to challenge the capacity of the optimizer with regard to different aspects. For example, WFG4 is a separable and strongly multi-modal problem while WFG5 is also a separable problem but it has a set of deceptive locally optimal fronts. WFG6 is also separable without the strong multi-modality of WFG4. The remaining three problems have the added difficulty of having a parameter-based bias. WFG7 is uni-modal and separable, like WFG4 and WFG6; WFG8 is a non-separable problem and WFG9 is non-separable, multi-modal and has deceptive local-optima. Each problem was configured with 3, 6, 9, 12 and 15 objective functions. For all cases the decision set dimension was fixed to 30.

Besides applying MARTEDA to the aforementioned problems some other MOEDAs and MOEAs are also assessed in order to provide a comparative ground for the tests. One algorithm is of particular interest, the MONEDA [37] algorithm. This approach was previously proposed by the authors to deal with the model-building issue of MOEDAs and MARTEDA is supposed to be an improvement over it. However, as MONEDA used the less-performing NSGA-II selection, we have also tested MONEDA with the HypE selection, in order to have some basis for comparison.

Besides MONEDA, we also tested the naïve MIDEA [11], MrBOA [1] and RM-RMEDA [57] MOEDAs and the SMS-EMOA [8], HypE [5], MOEA/D [55], and

NSGA-II [20] MOEAs. The parameters of the algorithms are summarized in Table 1. For each problem/dimension pair each algorithm was executed 30 times. The quality of the solutions is determined by the use of the hypervolume indicator [31].

Table 1 Parameters of the algorithms used in the experiments

Common parameters (if applicable)	
Population size (n_{pop})	$250 \cdot 10^{\frac{M}{3}-1}$
MARTEDA	
F2 vigilance threshold (ρ)	0.05
Initial standard deviations (φ)	0.01
Selection percentile (α)	0.3
\hat{P}_l to N^* ratio (γ)	0.5
Substitution percentile (ω)	0.25
MONEDA	
Number of initial GNG nodes (N_0)	2
Maximum edge age (v_{max})	40
Best node learning rate (ε_b)	0.1
Neighboring nodes learning rate (ε_v)	0.05
Insertion error decrement rate (δ_I)	0.1
General error decrement rate (δ_G)	0.1
Accumulated error threshold (ρ)	0.2
Selection percentile (α)	0.3
$\hat{\mathcal{P}}_l$ to N_{max} ratio (γ)	0.5
Substitution percentile (ω)	0.25
Naïve MIDEA	
Selection percentile (τ)	0.3
Diversity percentile (δ)	15
Number of parents of a variable (κ)	2
Maximum number of clusters	$\lceil 0.5 \lfloor \tau n_{\text{pop}} \rfloor \rceil$
Threshold for the leader algorithm	0.1
SMS-EMOA	
Distribution index for SBX (η_c)	15
Dist. index for polyn. mutation (η_m)	20
MrBOA	
Selection portion (τ)	0.3
Number of parents of a variable	5
Number of mixture components	3
Threshold of leader algorithm	0.3
MOEA/D	
Number of subproblems (N)	$300M$
Number of the weight vectors in the neighborhood of weight vector (T)	$0.1N$
Max. num. of solutions replaced (n_r)	$0.01N$
Parent solution selection prob. (δ)	0.9
NSGA-II	
Tournament selection size	5
Number of parent individuals	$\lfloor 0.3n_{\text{pop}} \rfloor$
Number of offspring individuals	$\lfloor 0.25n_{\text{pop}} \rfloor$
Crossover probability (p_c)	0.7
Distribution index for SBX (η_c)	15
Mutation probability (p_m)	$\frac{1}{n_{\text{pop}}}$
Dist. index for polyn. mutation (η_m)	20
Monte Carlo sample size (for $M > 3$)	10^4

Table 1 (continued)

RM-MEDA	
Selection portion	0.3
Number of LPCA clusters	$\frac{10}{3}M$
Maximum training steps in LPCA	$\frac{20}{3}M$
Extension rate	0.25
HypE	
—Variation operators as in NSGA-II—	
Monte Carlo sample size (for $M > 3$)	10^4

Note that each parameter follows the notation of the algorithm's author(s)

The stochastic nature of evolutionary algorithms prompts the use of statistical tools in order to reach a valid judgement of the quality of the solutions and how different algorithms compare with each other. Box plots [19] are one of such representations and have been repeatedly applied in our context. Although box plots allows a visual comparison of the results and, in principle, some conclusions could be deduced out of them.

Still, as we are aware that box plots are a helpful and popular visualization tool we have included the ones corresponding to our experiments in Figs. 3 and 4 for problems WFG4–WFG6 and WFG7–WFG9, respectively. Both sets of experiments share some interesting results. First of all, it is noticeable that in most problems, when configured with three objectives the results are very similar. In many cases, even when some algorithms perform better than others, the performance indicators of the different runs overlap. Probably this situation might change if the parameters were tuned for each problem configuration instead of keeping them constant for all instances. This situation changes dramatically as the number of objectives is increased.

Nevertheless, in order to reach a substantiated judgement it is necessary go beyond reporting the descriptive statistics of the performance indicators. For this task is required to carry out a set of statistical inferences that would support any judgements made from the data.

The statistical validity of the judgment of the results calls for the application of statistical hypothesis tests. It has been previously remarked by different authors that the Mann–Whitney–Wilcoxon U test [34] is particularly suited for experiments in the context of multi-objective evolutionary optimization [31]. This test is commonly used as a non-parametric method for testing equality of population medians. In our case we performed pair-wise tests on the significance of the difference of the indicator values yielded by the executions of the algorithms. A significance level, α , of 0.05 was used for all tests.

The visual analysis of the results is rather difficult as it implies cross-examining and comparing the results presented separately. That is why we decided to adopt a more integrative representation such as the one proposed in [3]. That is, for a given set of algorithms A_1, \dots, A_K , a set of P test problem instances $\Phi_{1,m}, \dots, \Phi_{P,m}$, configured with m objectives, the function $\delta(\cdot)$ is defined as

$$\delta(A_i, A_j, \Phi_{p,m}) = \begin{cases} 1 & \text{if } A_i \gg A_j \text{ solving } \Phi_{p,m} \\ 0 & \text{in other case} \end{cases}, \quad (20)$$

where the relation $A_i \gg A_j$ defines if A_i is significantly better than A_j when solving the problem instance $\Phi_{p,m}$, as computed by the statistical tests previously described.

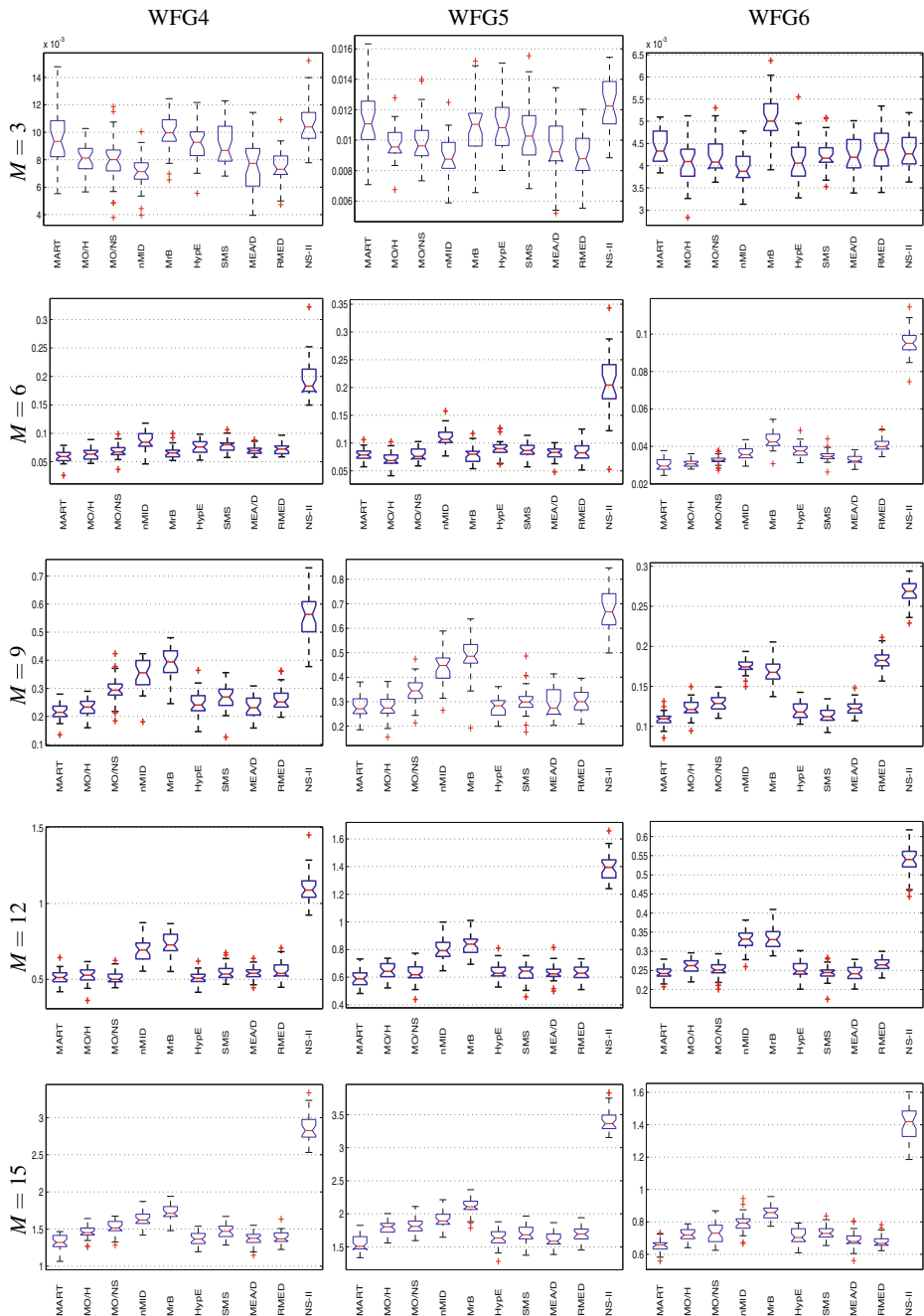


Fig. 3 Summary as box plots of hypervolume indicator obtained from the results of MARTEDA (MART), hypervolume-based MONEDA (MO/H), non-dominated sorting MONEDA (MO/NS), naïve MIDEA (nMID), MrBOA (MrB), HypE, SMS-EMOA (SMS), MOEA/D (MEA/D) and NSGA-II (NS-II) when solving WFG4, WFG5 and WFG6 with different number of objectives

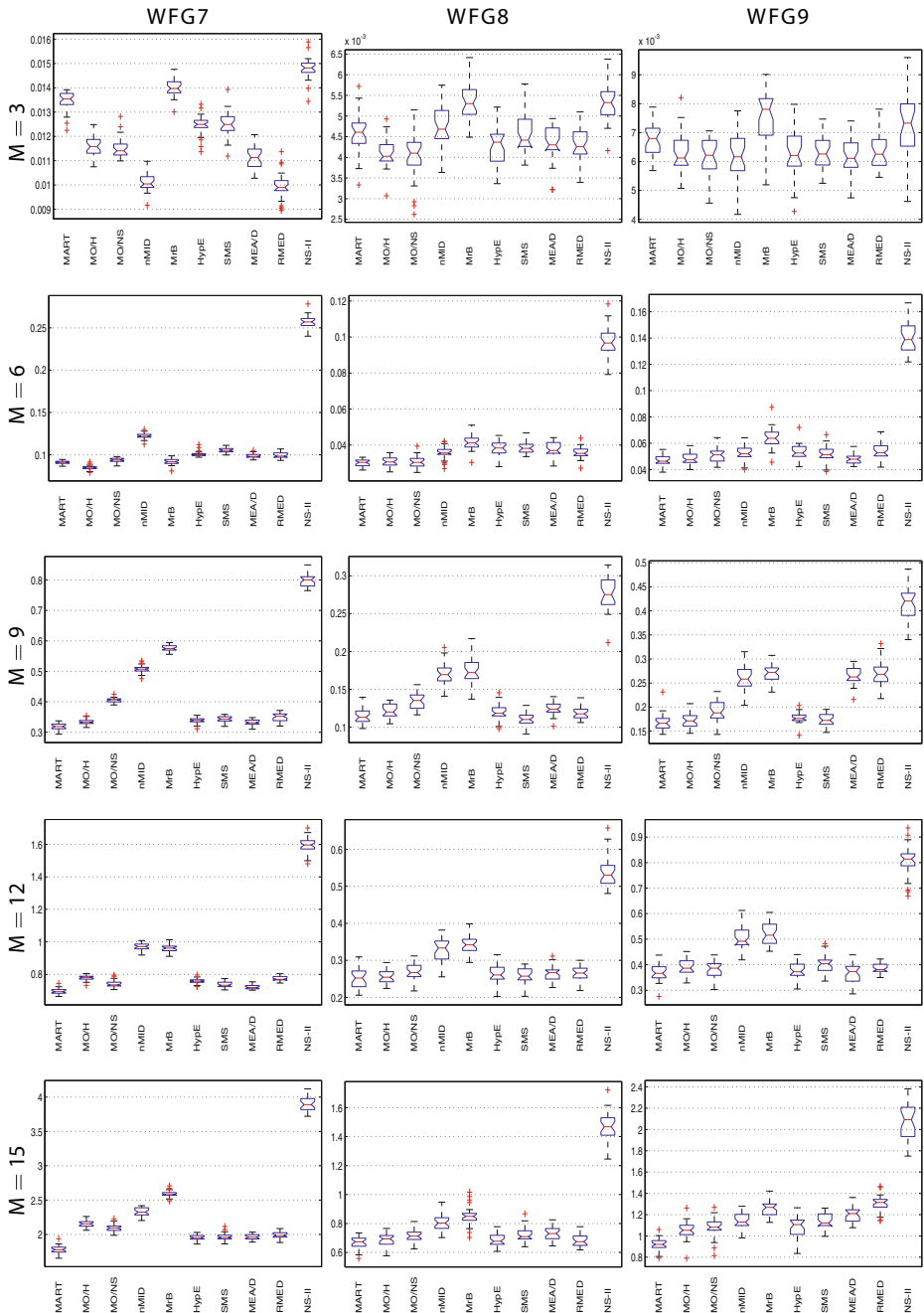


Fig. 4 Summary as box plots of hypervolume indicator obtained from the results of MARTEDA (MART), hypervolume-based MONEDA (MO/H), non-dominated sorting MONEDA (MO/NS), naïve MIDEA (nMID), MrBOA (MrB), HypE, SMS-EMOA (SMS), MOEA/D (MEA/D) and NSGA-II (NS-II) when solving WFG4, WFG5 and WFG6 with different number of objectives

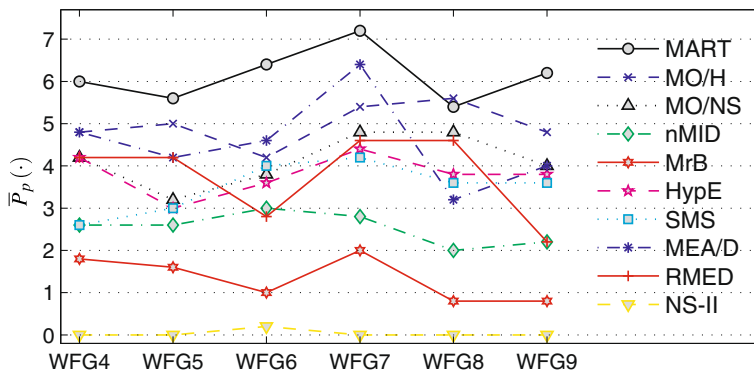


Fig. 5 Mean values of the performance index of MARTEDA (MART), MONEDA with HypE (MON/H) or NSGA-II selection (MON/NS), naïve MIDEA (n.MID), MrBOA, HypE, SMS-EMOA (SMS-EM), MOEA/D (MEA/D) and NSGA-II (NSG-II) across the different problems, $\bar{P}_p(\cdot)$

Relying on $\delta(\cdot)$, the performance index $P_{p,m}(A_i)$ of a given algorithm A_i when solving $\Phi_{p,m}$ is then computed as

$$P_{p,m}(A_i) = \sum_{j=1; j \neq i}^K \delta(A_i, A_j, \Phi_{p,m}). \quad (21)$$

This index intends to summarize the performance of each algorithm with regard to its peers.

Figures 5 and 6 exhibit the results computing the performance indexes grouped by problems and dimensions.

Figure 5 represents the mean performance indexes yielded by each algorithm when solving each problem in all of its configured objective dimensions,

$$\bar{P}_p(A_i) = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} P_{p,m}(A_i), \quad \text{with } \mathcal{M} = \{3, 6, 9, 12, 15\} \text{ in our case.} \quad (22)$$

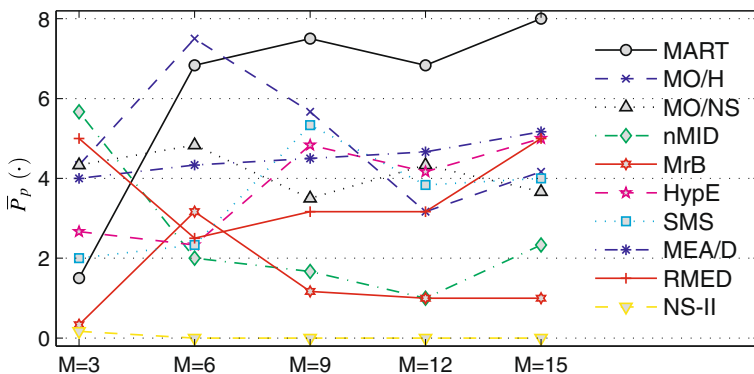


Fig. 6 Mean values of the performance index across the different space dimensions, $\bar{P}_m(\cdot)$. See Fig. 5 for a description of the acronyms

It is worth noticing that MARTEDA has better overall results with respect to the other algorithms in all problems, except WFG7. As it could be expected, the use of indicator-based selection in MONEDA has yielded better results than the original MONEDA. Indicator-based MONEDA and the indicator-based MOEAs have a similar performance. It can be hypothesized that these results can be biased by the three objective problems, having dramatic differences in their results with respect to the rest of the dimensions considered. In the case of WFG7, that is unimodal and separable, it could be argued that it poses a lesser challenge to the optimizers, and, in many cases, some of them could yield similar results to MARTEDA.

This situation is clarified in Fig. 6, which presents the mean values of the index computed for each dimension

$$\bar{P}_m(A_i) = \frac{1}{P} \sum_{p=1}^P P_{p,m}(A_i). \quad (23)$$

In this case, MARTEDA exhibits an outstanding performance with regard to the other algorithms in more than three dimensions. Still, another important conclusion can be extracted. For more than three objectives, the MOEDAs that attempt to tackle the model-building issue (MONEDA and MARTEDA) and that also exploit indicator-based selection have outperformed the rest of the methods. This is very important, as it transcends the particular results of a given algorithm but instead casts some light on what should be the proper trend of development in this field.

It is relevant the fact that, as the objective set dimension is increased, the difference of performance between MARTEDA and the rest becomes more evident.

Finally, the above experiments lead us to hypothesize that thanks to the treatment of the outliers in the model-building data-set, the MARTEDA approach manages to overcome the difficulties that hampers the rest of the methods. Another important result is that MARTEDA was able to yield good results across a varied set of problems without tuning its parameters in every case. This implies that MARTEDA has a certain degree of robustness regarding its parameters.

6.2 CEC'09 problems

As already commented, we will be now focusing on a set of problems previously proposed in the CEC 2009 MOP competition [58]. From that set of problems proposed we selected the unconstrained optimization problems UF1 to UF7. These are seven two-objective problems that can be configured to have any desired number of variables. These problems are well-known for the complexity of their Pareto-optimal sets and fronts. They were selected in order to be able to plot the results and to visually compare results.

The problems were configured with 30 decision variables and population size, when applicable, was set to 1,000 individuals. The rest of the experimental setup was kept as in the previous experiment.

Figure 7 contains the plots of the Pareto-optimal fronts produced by different algorithms. It is noticeable how in some cases algorithms converge to local Pareto-optimal fronts, not being able to progress towards the optimal solution.

This form of result presentation is valuable from a didactic point of view. However, in order to reach a substantiated judgement a methodology like the one described for the previous experiment should be followed. The outcome of applying

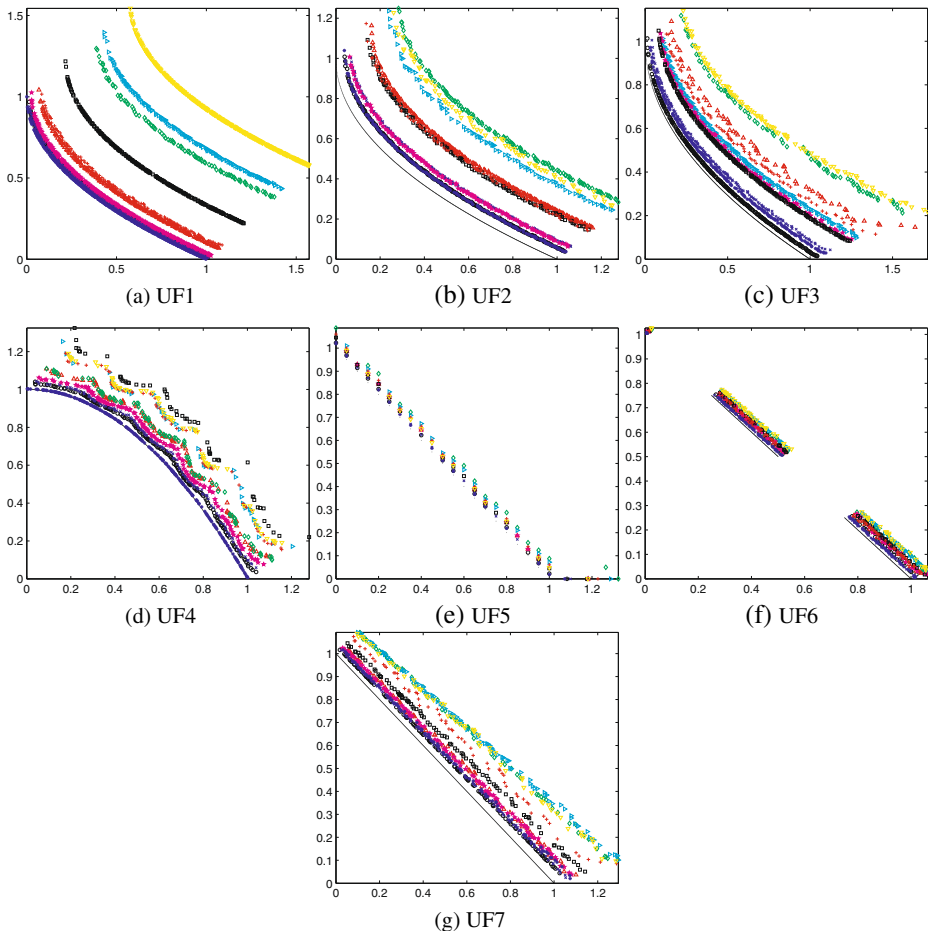


Fig. 7 Non-dominated fronts produced by MARTEDA (\circ), hypervolume-based MONEDA (\times), non-dominated sorting MONEDA (\triangle), naïve MIDEA (\diamond), MrBOA (\triangleright), HypE (\star), SMS-EMOA (\square), MOEA/D ($*$), RM-MEDA ($+$) and NSGA-II (∇) when solving UF1–UF7 problems. Pareto-optimal fronts (\cdot) are also shown

that approach to the experimental results can be observed as box plots in Fig. 8. Here it can be noted that MARTEDA yielded remarkable results, competing in many cases with MOEA/D for the best position.

Figure 9 summarizes the results of the experiments. This figure prompts a very interesting conclusion. In most problems it can be observed that there is an established difference on the performance of the algorithms. However, in UF5 many algorithms yield relatively similar results. This might be an indication that either this is a more complex problems or that more resources should be allotted to the optimization. In any case, the outperformance of MARTEDA can be asserted as it is one of the best performing methods.

It is noticeable, however that also MOEA/D and RM-MEDA manage to yield quality results. Methods based on regularity, like RM-MEDA, exploit the properties (those required to apply the KKT condition) of the problem being solved. In real

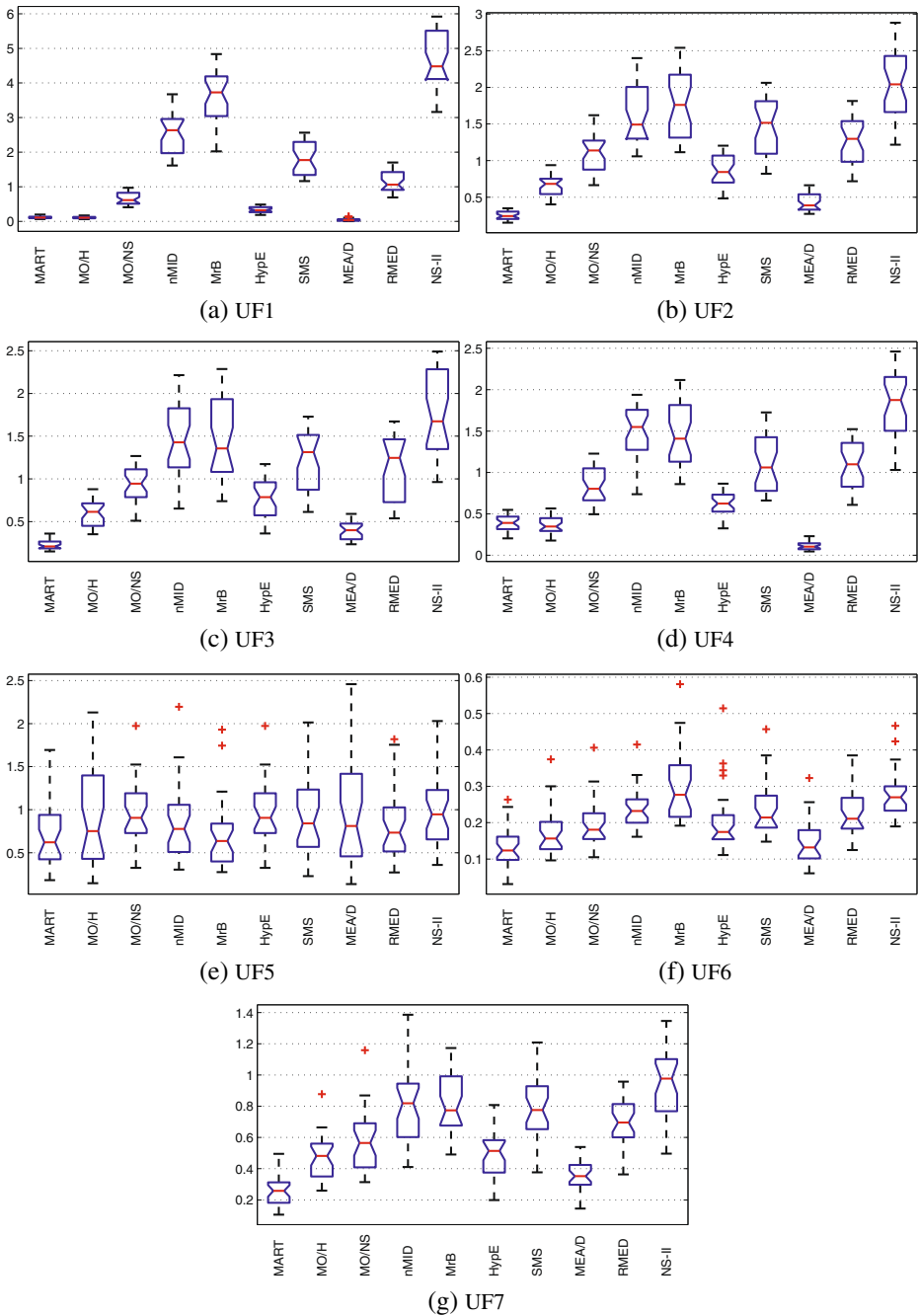


Fig. 8 Summary as box plots of hypervolume indicator obtained from the results of MARTEDA (MART), hypervolume-based MONEDA (MO/H), non-dominated sorting MONEDA (MO/NS), naïve MIDEA (nMID), MrBOA (MrB), HypE, SMS-EMOA (SMS), MOEA/D (MEA/D), RM-MEDA (RMED) and NSGA-II (NS-II) when solving problems UF1 to UF7

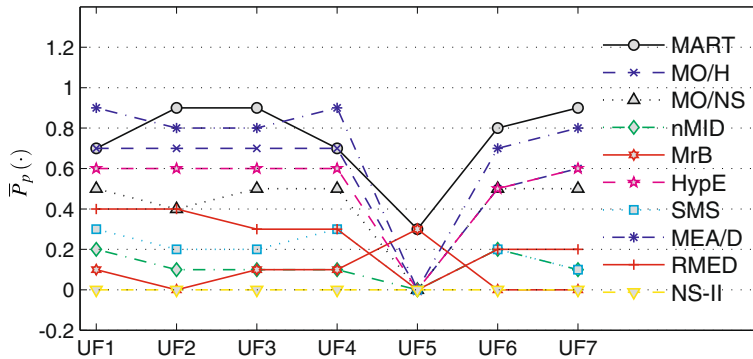


Fig. 9 Mean values of the performance index of MARTEDA (MART), MONEDA with HypE (MON/H) or NSGA-II selection (MON/NS), naïve MIDEA (n.MID), MrBOA, HypE, SMS-EMOA (SMS-EM), MOEA/D (MEA/D) and NSGA-II (NSG-II) across the different problems, $\bar{P}_p(\cdot)$

life practice the properties it is often impossible to check if the problem meets the requirements to apply the KKT condition. Furthermore, as they model the Pareto front as an $(m - 1)$ -dimensional manifold they have high computational demands as the number of objectives grows and these many-objective problems are the ones in which we are more interested.

7 Final remarks

In this paper we have explored the model-building issue of MOEDAs and the requirements it imposes on its supporting learning paradigm. We put forward adaptive resonance theory as a alternative learning paradigm. Based on it, we introduced a novel algorithm called multi-objective ART-based EDA (MARTEDA) that uses a Gaussian ART neural network for model-building and the hypervolume-based selection described for the HypE algorithm. We showed that by using this novel model-building approach and an indicator-based population ranking the algorithm is able to outperform similar MOEDAs and MOEAs.

Still, the main conclusion of this work is that we provide strong evidences that further research must be dedicated to the model-building issue in order to make current MOEDAs capable of dealing with complex multi-objective problems with many objectives. In spite of the fact that obviously further studies are necessary, these extensive experiments have provided solid ground for the use of MARTEDA in a real-world application context.

From a theoretical perspective some points still need to be explored. For example, a computational complexity study is necessary in order to grasp the resource consumption of MARTEDA when advancing into higher dimensions. On the other hand, more model-building approaches that follow our premises should be tested. Getting a proper understanding of the implication of following the learning paradigm put forward by ART in the context of model building, in particular from an statistical point of view is also necessary. If new ART-based approaches show themselves

as efficient as the one described here, it would help to corroborate the working hypothesis of this contribution.

Acknowledgements This work was supported by projects Projects CICYT TIN2011-28620-C02-01, CICYT TEC2011-28626-C02-02, CAM CONTEXTS (S2009/TIC-1485) and DPS2008-07029-C02-02.pt

References

1. Ahn, C.W.: *Advances in Evolutionary Algorithms. Theory, Design and Practice*. Springer (2006). ISBN 3-540-31758-9
2. Bäck, T.: *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York (1996)
3. Bader, J.: *Hypervolume-based search for multiobjective optimization: theory and methods*. Ph.D. thesis, ETH Zurich, Switzerland (2010)
4. Bader, J., Deb, K., Zitzler, E.: Faster hypervolume-based search using Monte Carlo sampling. In: Beckmann, M., Küenzi, H.P., Fandel, G., Trockel, W., Basile, A., Drexler, A., Dawid, H., Inderfurth, K., Kürsten, W., Schittko, U., Ehrgott, M., Naujoks, B., Stewart, T.J., Wallenius, J. (eds.) *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems. Lecture Notes in Economics and Mathematical Systems*, vol. 634, pp. 313–326. Springer, Berlin (2010). doi:[10.1007/978-3-642-04045-0_27](https://doi.org/10.1007/978-3-642-04045-0_27)
5. Bader, J., Zitzler, E.: HypE: an algorithm for fast hypervolume-based many-objective optimization. *Evol. Comput.* **19**(1), 45–76 (2011). doi:[10.1162/EVCO_a_00009](https://doi.org/10.1162/EVCO_a_00009), URL: http://www.mitpressjournals.org/doi/abs/10.1162/EVCO_a_00009
6. Bellman, R.E.: *Adaptive Control Processes*. Princeton University Press, Princeton (1961)
7. Beume, N.: S-metric calculation by considering dominated hypervolume as Klee's measure problem. *Evol. Comput.* **17**(4), 477–492 (2009). doi:[10.1162/evco.2009.17.4.17402](https://doi.org/10.1162/evco.2009.17.4.17402), URL: <http://www.mitpressjournals.org/doi/abs/10.1162/evco.2009.17.4.17402>, PMID: 19916778
8. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: multiobjective selection based on dominated hypervolume. *Eur. J. Oper. Res.* **181**(3), 1653–1669 (2007). URL: <http://ideas.repec.org/a/eee/ejores/v181y2007i3p1653-1669.html>
9. Beume, N., Rudolph, G.: Faster S-metric calculation by considering dominated hypervolume as Klee's measure problem. In: Kovalerchuk, B. (ed.) *Proceedings of the Second IASTED International Conference on Computational Intelligence*, pp. 233–238. IASTED/ACTA Press (2006)
10. Bosman, P.A.N.: *Design and application of iterated density-estimation evolutionary algorithms*. Ph.D. thesis, Institute of Information and Computing Sciences, Universiteit Utrecht, Utrecht, The Netherlands (2003)
11. Bosman, P.A.N., Thierens, D.: The naïve MIDEA: A baseline multi-objective EA. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*. Springer, Lecture Notes in Computer Science, vol. 3410, pp. 428–442. Guanajuato, México (2005)
12. Box, G.E.P., Muller, M.E.: A note on the generation of random normal deviates. *Ann. Math. Stat.* **29**, 610–611 (1958)
13. Branke, J., Lode, C., Shapiro, J.L.: Addressing sampling errors and diversity loss in umda. In: *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation—GECCO '07*, pp. 508–515. ACM Press (2007). doi:[10.1145/1276958.1277068](https://doi.org/10.1145/1276958.1277068), URL: <http://portal.acm.org/citation.cfm?doid=1276958.1277068>
14. Branke, J., Miettinen, K., Deb, K., Słowiński, R. (eds.): *Multiobjective Optimization. Lecture Notes in Computer Science*, vol. 5252. Springer, Berlin (2008)
15. Bringmann, K., Friedrich, T.: Approximating the volume of unions and intersections of high-dimensional geometric objects. *Comput. Geom.* **43**(6–7), 601–610 (2010). doi:[10.1016/j.comgeo.2010.03.004](https://doi.org/10.1016/j.comgeo.2010.03.004), URL: <http://linkinghub.elsevier.com/retrieve/pii/S0925772110000167>
16. Brockhoff, D., Saxena, D.K., Deb, K., Zitzler, E.: On handling a large number of objectives a posteriori and during optimization. In: Knowles, J., Corne, D., Deb, K. (eds.) *Multi-Objective Problem Solving from Nature: From Concepts to Applications*, Natural Computing Series, pp. 377–403. Springer (2008). doi:[10.1007/978-3-540-72964-8](https://doi.org/10.1007/978-3-540-72964-8)

17. Brockhoff, D., Zitzler, E.: Dimensionality reduction in multiobjective optimization: the minimum objective subset problem. In: Waldmann, K.H., Stocker, U.M. (eds.) *Operations Research Proceedings 2006*, pp. 423–429. Springer (2007)
18. Carpenter, G., Grossberg, S.: The ART of adaptive pattern recognition by a self-organizing neural network. *Computer* **21**(3), 77–88 (1988). doi:[10.1109/2.33](https://doi.org/10.1109/2.33)
19. Chambers, J., Cleveland, W., Kleiner, B., Tukey, P.: *Graphical Methods for Data Analysis*. Wadsworth, Belmont (1983)
20. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd edn. Genetic and Evolutionary Computation. Springer, New York (2007). URL: <http://www.springer.com/west/home/computer/foundations?SGWID=4-156-22-173660344-0>
21. Corne, D.W.: Single objective = past, multiobjective = present, ??? = future. In: Michalewicz, Z. (ed.) *2008 IEEE Conference on Evolutionary Computation (CEC)*, Part of 2008 IEEE World Congress on Computational Intelligence (WCCI 2008). IEEE Press, Piscataway (2008). doi:[10.1109/CEC.2008.4631179](https://doi.org/10.1109/CEC.2008.4631179)
22. Deb, K.: *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, Chichester (2001). ISBN 0-471-87339-X
23. Deb, K., Saxena, D.K.: Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In: *2006 IEEE Conference on Evolutionary Computation (CEC'2006)*, pp. 3352–3360. IEEE Press, Piscataway (2006)
24. Deolalikar, V.: $P \neq NP$. Tech. rep., Hewlett Packard Research Labs, Palo Alto, CA, USA (2010). URL: <http://www.scribd.com/doc/35539144/pnp12pt>
25. Fleischer, M.: The measure of Pareto optima. Applications to multi-objective metaheuristics. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) *Evolutionary Multi-Criterion Optimization*. Second International Conference, EMO 2003. Springer. Lecture Notes in Computer Science, vol. 2632, pp. 519–533. Faro, Portugal (2003)
26. Fonseca, C.M., Paquete, L., López-Ibáñez, M.: An improved dimension-sweep algorithm for the hypervolume indicator. In: *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pp. 1157–1163 (2006)
27. Grossberg, S.: How does the brain build a cognitive code? *Psychol. Rev.* **87**, 1–51 (1980)
28. Grossberg, S.: *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control*. Reidel, Boston (1982)
29. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Trans. Evol. Comput.* **10**(5), 477–506 (2006)
30. Khare, V., Yao, X., Deb, K.: Performance scaling of multi-objective evolutionary algorithms. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) *Evolutionary Multi-Criterion Optimization*. Second International Conference, EMO 2003. Springer. Lecture Notes in Computer Science, vol. 2632, pp. 376–390. Faro, Portugal (2003)
31. 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)
32. Knowles, J.D.: *Local-search and hybrid evolutionary algorithms for Pareto optimization*. Ph.D. thesis, The University of Reading, Department of Computer Science, Reading, UK (2002)
33. Lozano, J.A., Larrañaga, P., Inza, I., Bengoetxea, E. (eds.): *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*. Springer (2006)
34. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. *Ann. Math. Stat.* **18**, 50–60 (1947)
35. Martí, L., García, J., Berlanga, A., Coello Coello, C.A., Molina, J.M.: On current model-building methods for multi-objective estimation of distribution algorithms: shortcomings and directions for improvement. Tech. Rep. GIAA2010E001, Grupo de Inteligencia Artificial Aplicada, Universidad Carlos III de Madrid, Colmenarejo, Spain (2010). URL: <http://www.giaa.inf.uc3m.es/miembros/lmarti/model-building>
36. Martí, L., García, J., Berlanga, A., Coello Coello, C.A., Molina, J.M.: MB-GNG: Addressing drawbacks in multi-objective optimization estimation of distribution algorithms. *Oper. Res. Lett.* **39**(2), 150–154 (2011). doi:[10.1016/j.orl.2011.01.002](https://doi.org/10.1016/j.orl.2011.01.002)
37. Martí, L., García, J., Berlanga, A., Molina, J.M.: Introducing MONEDA: scalable multiobjective optimization with a neural estimation of distribution algorithm. In: Keizer, M., Antoniol, G., Congdon, C., Deb, K., Doerr, B., Hansen, N., Holmes, J., Hornby, G., Howard, D., Kennedy, J., Kumar, S., Lobo, F., Miller, J., Moore, J., Neumann, F., Pelikan, M., Pollack, J., Sastry, K., Stanley, K., Stoica, A., Talbi, E.G., Wegener, I. (eds.) *GECCO '08: 10th Annual*

- Conference on Genetic and Evolutionary Computation, pp. 689–696. ACM Press, New York (2008). doi:[10.1145/1389095.1389230](https://doi.org/10.1145/1389095.1389230). EMO Track “Best Paper” Nominee
38. Michalski, R.S.: Learnable evolution model: evolutionary processes guided by machine learning. *Mach. Learn.* **38**, 9–40 (2000)
 39. Papadimitriou, C.M.: Computational Complexity. Addison-Wesley, Reading (1994)
 40. 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)
 41. Pelikan, M., Sastry, K., Goldberg, D.E.: Multiobjective estimation of distribution algorithms. In: Pelikan, M., Sastry, K., Cantú-Paz, E. (eds.) *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*, Studies in Computational Intelligence, pp. 223–248. Springer (2006)
 42. Pena, J.M., Robles, V., Larrañaga, P., Herves, V., Rosales, F., Pérez, M.S.: GA-EDA: hybrid evolutionary algorithm using genetic and estimation of distribution algorithms. In: *Innovations in Applied Artificial Intelligence*, pp. 361–371. Springer, Heidelberg (2004). URL: <http://www.springerlink.com/index/237R7QCABF08NT98.pdf>
 43. Praditwong, K., Yao, X.: How well do multi-objective evolutionary algorithms scale to large problems. In: 2007 IEEE Congress on Evolutionary Computation (CEC 2007), pp. 3959–3966. IEEE Press, Piscataway (2007). doi:[10.1109/CEC.2007.4424987](https://doi.org/10.1109/CEC.2007.4424987)
 44. Purshouse, R.C., Fleming, P.J.: On the evolutionary optimization of many conflicting objectives. *IEEE Trans. Evol. Comput.* **11**(6), 770–784 (2007). doi:[10.1109/TEVC.2007.910138](https://doi.org/10.1109/TEVC.2007.910138)
 45. Sarle, W.S.: Why statisticians should not FART. Tech. rep., SAS Institute, Cary, NC (1995)
 46. Sheri, G., Corne, D.W.: The simplest evolution/learning hybrid: LEM with KNN. In: IEEE World Congress on Computational Intelligence, pp. 3244–3251. IEEE Press, Hong Kong (2008)
 47. Sheri, G., Corne, D.W.: Learning-assisted evolutionary search for scalable function optimization: LEM(ID3). In: IEEE World Congress on Computational Intelligence. IEEE Press, Barcelona (2010)
 48. Stewart, T.J., Bandte, O., Braun, H., Chakraborti, N., Ehrgott, M., Göbelt, M., Jin, Y., Nakayama, H., Poles, S., Di Stefano, D.: Real-world applications of multiobjective optimization. In: Branke, J., Miettinen, K., Deb, K., Słowiński, R. (eds.) *Multiobjective Optimization. Lecture Notes in Computer Science*, vol. 5252, pp. 285–327. Springer, Berlin (2008)
 49. Wagner, T., Beume, N., Naujoks, B.: Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) *4th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*. Lecture Notes in Computer Science, vol. 4403, pp. 742–756. Springer, Heidelberg (2007). doi:[10.1007/978-3-540-70928-2](https://doi.org/10.1007/978-3-540-70928-2), URL: <http://www.springerlink.com/index/10.1007/978-3-540-70928-2>
 50. Wagner, T., Beume, N., Naujoks, B.: Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) *Evolutionary Multi-Criterion Optimization (EMO 2007)*. Lecture Notes in Computer Science, vol. 4403, pp. 742–756. Springer, Berlin (2007). doi:[10.1007/978-3-540-70928-2_56](https://doi.org/10.1007/978-3-540-70928-2_56)
 51. While, L., Hingston, P., Barone, L., Huband, S.: A faster algorithm for calculating hypervolume. *IEEE Trans. Evol. Comput.* **10**(1), 29–38 (2006). doi:[10.1109/TEVC.2005.851275](https://doi.org/10.1109/TEVC.2005.851275)
 52. Williamson, J.R.: Gaussian ARTMAP: a neural network for fast incremental learning of noisy multidimensional maps. *Neural Netw.* **9**, 881–897 (1996)
 53. Williamson, J.R.: A constructive, incremental-learning network for mixture modeling and classification. *Neural Comput.* **9**, 1517–1543 (1997)
 54. Yuan, B., Gallagher, M.: On the importance of diversity maintenance in estimation of distribution algorithms. In: GECCO '05: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, pp. 719–726. ACM Press, New York (2005). doi:[10.1145/1068009.1068129](https://doi.org/10.1145/1068009.1068129)
 55. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **11**(6), 712–731 (2007). doi:[10.1109/TEVC.2007.892759](https://doi.org/10.1109/TEVC.2007.892759)
 56. Zhang, Q., Sun, J., Tsang, E.: An evolutionary algorithm with guided mutation for the maximum clique problem. *IEEE Trans. Evol. Comput.* **9**(2), 192–200 (2005). doi:[10.1109/TEVC.2004.840835](https://doi.org/10.1109/TEVC.2004.840835), URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1413259>
 57. Zhang, Q., Zhou, A., Jin, Y.: RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm. *IEEE Trans. Evol. Comput.* **12**(1), 41–63 (2008). doi:[10.1109/TEVC.2007.894202](https://doi.org/10.1109/TEVC.2007.894202)
 58. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P., Liu, W., Tiwari, S.: Multiobjective optimization test instances for the CEC 2009 special session and competition. Tech. rep., University of Essex, Colchester, UK and Nanyang Technological University, Singapore (2009)

59. Zitzler, E., Brockhoff, D., Thiele, L.: The hypervolume indicator revisited: on the design of pareto-compliant indicators via weighted integration. In: Obayashi, S., et al. (eds.) *Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*. LNCS, vol. 4403, pp. 862–876. Springer, Berlin (2007)
60. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms on test functions of different difficulty. In: Wu, A.S. (ed.) *Proceedings of the 1999 Genetic and Evolutionary Computation Conference. Workshop Program*, pp. 121–122. Orlando, Florida (1999)
61. Zitzler, E., Laumanns, M., Thiele, L., Fonseca, C.M., Grunert da Fonseca, V.: Why quality assessment of multiobjective optimizers is difficult. In: Langdon, W.B., Cantú-Paz, E., Mathias, K., Roy, R., Davis, D., Poli, R., Balakrishnan, K., Honavar, V., Rudolph, G., Wegener, J., Bull, L., Potter, M., Schultz, A., Miller, J., Burke, E., Jonoska, N. (eds.) *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002)*, pp. 666–673. Morgan Kaufmann, San Francisco (2002)
62. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans. Evol. Comput.* **7**(2), 117–132 (2003)