Applied Soft Computing 36 (2015) 87–100

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A surrogate-assisted evolutionary algorithm based on the genetic diversity objective

Andrea Massaro, Ernesto Benini *

Department of Industrial Engineering of the University of Padova, Via Venezia, 1, 35131 Padova, Italy

ARTICLE INFO

Article history:
Received 23 December 2011
Received in revised form 21 May 2015
Accepted 19 June 2015
Available online 17 July 2015

Keywords:
Surrogate models
Metamodels
Memetic algorithms
Genetic diversity
Evolutionary algorithm design

ABSTRACT

In this work, a novel surrogate-assisted memetic algorithm is proposed which is based on the preservation of genetic diversity within the population. The aim of the algorithm is to solve multi-objective optimization problems featuring computationally expensive fitness functions in an efficient manner. The main novelty is the use of an evolutionary algorithm as global searcher that treats the genetic diversity as an objective during the evolution and uses it, together with a non-dominated sorting approach, to assign the ranks. This algorithm, coupled with a gradient-based algorithm as local searcher and a back-propagation neural network as global surrogate model, demonstrates to provide a reliable and effective balance between exploration and exploitation. A detailed performance analysis has been conducted on five commonly used multi-objective problems, each one involving distinct features that can make the convergence difficult toward the Pareto-optimal front. In most cases, the proposed algorithm outperformed the other state-of-the-art evolutionary algorithms considered in the comparison, assuring higher repeatability on the final non-dominated set, deeper convergence level and higher convergence rate. It also demonstrates a clear ability to widely cover the Pareto-optimal front with larger percentage of non-dominated solutions if compared to the total number of function evaluations.

1. Introduction

In recent years, the research toward building advanced multi-objective evolutionary algorithms (MOEAs) for solving complex problems involving multiple conflicting objectives have been increasing enormously [1]. In particular, the need for effective optimization tools dealing with computationally expensive objective functions and constraints has become widespread in almost all the engineering disciplines. One of these is computational fluid dynamics (CFD), a set of numerical techniques which makes it possible to solve for the Reynolds-averaged Navier–Stokes (RANS) equations in complex domains, where a single run can require a huge computational effort.

In those cases, the modeling and design optimization cycle time is roughly proportional to the number of calls to the computationally expensive solver, so that many evolutionary frameworks have been implemented around the idea of alleviating the computational cost by introducing an approximate, or surrogate, model of the real objective functions [2,3]. The surrogate model (SM) is actually a model of the mathematical model itself upon which the solver is built, and therefore con be referred to as a "metamodel".

Using a SM has become a very popular approach since the efforts required to build the surrogates and to use them along with MOEAs are much lower than those in the standard, direct-call methodologies [4]. Popular SM are built using several methodologies, e.g. parametric statistical methods such as the response surface methodology (RSM) [5–7] as well as nonparametric techniques [8–10], like multivariate adaptive regression splines [11], artificial neural networks (ANNs) [12] and radial basis functions (RBFs) [13–15]. Other, more recently developed, nonparametric techniques for surrogate modeling encompass support vector regression (SVR) [16–18], regression Kriging (RK) [19–22] and moving least squares or local polynomial regression (LPR) [23–27].

In the framework of MOEAs, several approaches for dealing with computationally expensive problems using surrogate models have been documented in the open literature. First approaches were based on the concept of fitness inheritance [28,29], where the fitness of an individual is evaluated indirectly by interpolating the fitness of its parents. Evolution control techniques, e.g. clustering methods [30,31], were also proposed as a way to estimate when exact function evaluations are to be performed.

Alternative approaches utilize a progressive refinement of a SM as the search evolves [32]. Within those methodologies, memetic

* Corresponding author. Tel.: +39 049 8276767; fax: +39 049 8276785.
E-mail address: ernesto.benini@unipd.it (E. Benini).

http://dx.doi.org/10.1016/j.asoc.2015.06.026
1568-4946/

algorithms (MAs) seem to be one of the most promising techniques [33].

MAs are population-based metaheuristic search methods that follow the basis of Dawkins notion of "meme", defined as a unit of cultural evolution that is capable of local refinements [34–36]. MAs have also been used under the name of hybrid evolutionary algorithms, Baldwinian evolutionary algorithms, Lamarkian evolutionary algorithms, or genetic local search [33]. The main advantage of MAs over concurrent strategies lies in creating a synergy between global and local search of a set of optimal solutions of the objective function. A SM can be used in lieu of the original, computationally expensive, objective function during the local refinement thus leading to the so-called surrogate-assisted memetic algorithm (SAMA) concept. To this purpose, the prediction accuracy of fitness predictions based on SMs can be significantly enhanced with the inclusion of gradient information in SM building [37,38].

It has been extensively demonstrated in literature [39–41] that diversity preservation during evolution is a crucial aspect for evolutionary algorithms (EAs). If the lack of population diversity occurs too early the algorithm is trapped in confined regions and is not able to explore the whole search space. The diversity preservation becomes particularly important in the case of MAs [42,43] and SAMAs because of their marked tendency to exploitation. In this paper, a novel multi-objective SAMA is introduced and described which is based on the coupling between the genetic diversity evolutionary algorithm (GeDEA) [39] as global searcher and a local search framework built around a gradient-based algorithm working with a SM of the original optimization function. Because of its peculiar features we named it the genetic diversity memetic algorithm (GDMA).

In the following chapters the rationale behind the development of the proposed algorithm is presented, starting with a brief recall about the importance of preserving diversity during evolution for genetic and memetic algorithms. The main methodologies used in literature to face such a problem are listed and compared with the selected diversity preservation method [39]. Second the GDMA is presented together with the metamodeling technique used to build the local search framework, and the framework itself. In the final stage GDMA is applied to five, commonly used, test functions in order to compare its performance against several well-known evolutionary algorithms by means of four different performance metrics. GDMA is then used on a real-world application where a multi-objective/multi-point optimization of a 2D aerodynamic airfoil is carried out.

2. The preservation of genetic diversity

The genetic diversity of individuals for population-based algorithms has been recognized as a crucial property since the beginning of the subject [44] and many works were conducted with the aim of improving the performance of an algorithm by introducing specific diversity preservation strategies. As already mentioned the selective pressure driving toward optimal solutions can lead to a rapid impoverishment of the genetic material within the population, with consequent premature convergence of the algorithm. The strategies used in literature to prevent such a circumstance in GAs are many, in the following we briefly recall and describe the most relevant ones:

- Crowding: it consists on the replacement of existing individuals on the parent population that present similarities on a genotypic viewpoint. The strategies differ for the way they select and replace the most similar elements inside the population. Deterministic [40,45] and probabilistic [46] crowding methods have been used. Other crowding algorithms are Metropolis algorithm [47], restricted tournament selection [48] and simulated annealing [49], which can be all considered local tournament algorithms [50].
- Niching: these techniques consist on promoting the formation within the population of stable sub-populations (niches) [51] which contain different genetic information. They are traditionally used in domains when the finding of multiple solutions is desired, such as in optimization of multi-modal functions. The most frequently used niching technique is the fitness sharing [52].
- Diversity as an objective: the idea of using the diversity as an objective during the evolution was first in [39]. The diversity, measured in terms of the Euclidean distance from the other individuals, is used in the ranking procedure thanks to a non-dominated sorting of (i) the diversity and (ii) the ranks scored with respect to the objectives of the original MOOP. In [53,54] instead the diversity is used as an additional objective, increasing the dimensionality of the original MOOP.
- Other methods: additional methods are available in literature like multiploidy [55], DCGA (diversity control oriented GA) [56], CSGA (complementary surrogate GA) [57], TMPGA (Tabu multi parent GA) [58], FUSS (fitness uniform selection scheme) [59] and other strategies not cited here.

The importance of diversity has been highlighted also in some works where basic MAs (without the use of SMs) were used. In [60,61] the diversity has been used during the evolution of a parallel memetic algorithm (PMA) in order to dynamically control the local search frequency with the aim of reducing the number of function evaluations. On a similar fashion the individuals' diversity has been considered in [62] with the only difference that here it is used to decide between three different local searchers of an adaptive MA. The works considered so far treat the diversity as an indirect parameter to adapt the local search strategy, which is not strictly a diversity preservation technique. On the contrary in [63] a MA is improved by direct population management removing solutions that are below a certain threshold in terms of distance from the other individuals. Again in [42] the population diversity is preserved in a bacterial MA by means of hibernation of individuals. The idea is to hibernate some bacteria for a while. Then the hibernated bacteria can help in directing back the evolutionary process to right way if it convergences to local optima. No works are known at the moment of writing about diversity preservation techniques implemented in SAMAs.

In previous paragraph we stated the significance of preserving diversity within the evolutionary process in EAs and this is confirmed by the great importance that this subject has had in literature and the large number of works done on it. Diversity is considered as an important factor for classic GAs but it becomes crucial for MAs and SAMAs. These types of algorithms are specifically designed to emphasize the selective pressure mechanism toward the Pareto optimal set, thanks to the local search framework embedded. For this reason they are much more subject to premature convergence and impoverishment of the genotypic diversity within the population and a diversity preservation strategy is mandatory.

3. The genetic diversity memetic algorithm (GDMA)

The proposed optimization strategy consists in a SAMA strategy which is based on the genetic diversity preservation method GeDEA [39]. The GeDEA is here used as global searcher while additional tools are selected in order to set up a SAMA methodology. An ANN is selected as objective function's surrogate model and a Local Search Framework is built to perform local refinements of the

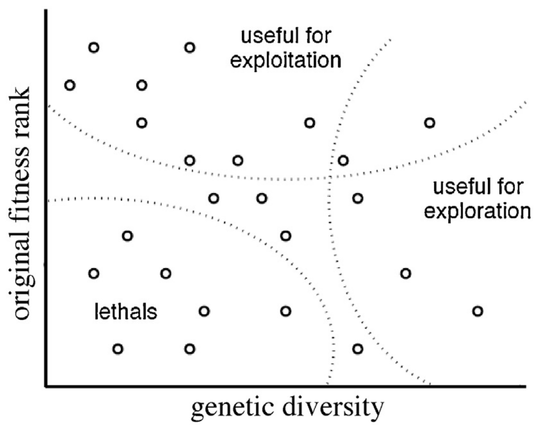


Fig. 1. GeDEM fitness assignment.

individuals using the aforementioned metamodel. The main tools and their implementation are described in detail below.

3.1. The GeDEA global searcher

GeDEA is a multi-objective evolutionary algorithm based on the preservation of the diversity among the evolution of the individuals. The basic idea behind the methodology, the Genetic Diversity Evaluation Method (GeDEM), consists in treating the diversity as a driving objective during the evaluation phase, emphasizing the non-dominated solutions as well as the most genetically different. This results in a selection pressure driving the search simultaneously toward the exploitation of the current non-dominated solutions and to the exploration of the search space. Fig. 1 shows the GeDEM's way of ranking the solutions.

The ranks of the individuals are determined maximizing the original ranks scored with respect to the objectives of the original MOOP (useful for exploitation) and the values assigned to each individual as a measure of its genetic diversity (useful for exploration), calculated according to the chosen distance metric. In order to assign fitness to the individuals the GeDEM performs non-dominated sorting procedure among the two aforementioned objectives.

GeDEA is here used as global searcher within the SAMA framework because of its ability in preserving the diversity as the population evolves, thus counterbalancing the peculiar tendency to exploitation of MAs.

3.2. The artificial neural network (ANN) as global surrogate model

A surrogate-assisted strategy (for MAs) uses one or more surrogate models to perform a local refinement of the individuals that the global searcher (GeDEA in this case) provides. Several mathematical metamodels [65] are available to represent a n-dimensional function. In present work an Artificial Neural Network (ANN) is selected [66], since the following reasons:

- It yields better approximations compared to the classical response surface methods when the nature of the problem is unknown, even for discontinuous functions;
- It can easily manage a large number of design parameters;
- It shows good responses if the boundaries of the design space are not well-defined;
- It has nominally unlimited representation capabilities of the function's complexity.

A generic ANN is composed of basic elements, called neurons. Fig. 2 shows a neuron structure: a vector input p, with a number Q of elements, is multiplied by the weights w and the solution is added to the bias b. The result is used as an input for the transfer function f, that provides the neuron's output a. The transfer function f can be of different nature, the most commonly used for multi-layer networks are the Tan-Sigmoid and the Linear Transfer Function.

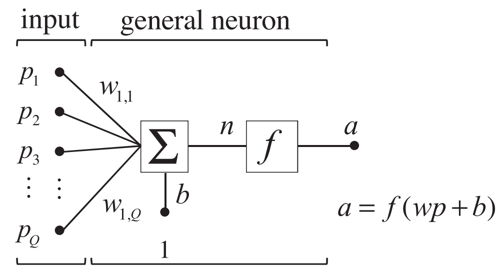


Fig. 2. General neuron architecture with input vector.

The neurons are connected together by a structure of links, as in a real biological nervous system. In an ANN the neurons are organized by layers with a finite number of them within each layer and the connections (weights), together with the biases, can vary their values to modify the response of the network. To reproduce a generic objective function the ANN requires a training, in which a set of starting individuals (stored in a database) are considered in relation with their scores. An iterative procedure changes the weights and the biases of the network until a sufficient approximation accuracy of the output data is reached.

The original function is here represented as a global model, therefore the ANN uses all the available individuals in the database in order to be trained and tries to follow the function along the whole variability range of the parameters. In this case the model is unique and represents the entire function. The specific ANN used here is a feed-forward neural network which is composed of an input, two hidden layers with respectively 10 and 16 Tan-Sigmoid neurons, and an output layer with 10 linear neurons.

Overfitting is one of the most important problems in ANN training, so to achieve a good generalization of the solution two techniques are employed. The first action to improve generalization is to use a network with the minimum number of neurons, just enough to adequate approximation. Larger networks can learn more complex problems but they have enough power to overfit the data. The second action is to use the available data to validate the network. The global database, containing individuals and scores from the true objective function evaluation, is divided into two subsets. The first one is the training subset, used to train the network. It contains the 75% of the data, that are chosen randomly. The remaining individuals composes the evaluation subset. The network is trained making use of the Levenberg–Marquardt algorithm (LM), a quasi-Newton method faster than the traditional back-propagation one. We use the mean square error (MSE) to measure the ANN performance, as defined in (1):

$$\mathrm{MSE} = \frac{1}{M_{TDB}} \sum_{i=1}^{M_{TDB}} (F_i - F_{NNi})^2 \tag{1}$$

where M_{TDB} is the number of available individuals in the training subset. The iterative training process is stopped when the algorithm reaches 100 epochs or when the training performance index MSE_{train} is lower than 5×10^{-7}. Finally, the evaluation performance index MSE_{eval} is calculated simulating the trained network on the evaluation subset. The training process is repeated more than once, in this case 3 times, because each training can evolve into a different model (since the random initialization of weights/biases and the random selection of the training subset) and only the surrogate model with the best (minimum) MSE_{eval} is accepted. The ANN model used here has been developed and adapted from [67].

3.3. The local search framework (LSF)

The local refinement of the GeDEA individuals is performed using the Local Search Framework (LSF). The proposed method for local search makes use of a single metamodel (the ANN in Section 3.2) and a single local searcher. The latter is based on an *active-set* algorithm [68,69] that is a sequential quadratic programming (SQP) method in which an estimate of the Hessian of the Lagrangian is updated at every iteration using the BFGS formula.

When the global searcher (GeDEA) is used for a multi-objective problem optimization the use of the local searcher (gradient-based algorithm) requires a specific treatment of the problem itself. Different strategies can be followed to handle this issue, like the conversion of the multi-objective problem into a weighted sum of objectives. This method lacks in generality and the selection of the weights by the user can be difficult, affecting the final results. A different technique is actually used here, which consists in the transformation of the multi-objective problem into a sequence of constrained single-objective problems. Each problem objective is minimized using the local searcher and its ANN approximation following a random sequence, while the remaining objectives are modified into inequality constraints. The new individual coming from the local refinement is accepted if, and only if, the selected objective has been improved respect to its original value and, contemporarily, the remaining objectives are lower than or equal to the original values.

3.4. The GDMA operation

The GDMA algorithm can be described by the following main steps:

Step 1: An initial number of generations using the GeDEA (GA) algorithm only, G_{GeDEA}, are performed, with a population size of μ individuals;
Step 2: First training of the ANN surrogate model, one for each objective, using the global set of individuals coming from the Step 1;
Step 3: The population from the GeDEA crossover (and mutation) is passed to the LSF framework. During the first SAMA generation the individuals are locally improved by the gradient-based algorithm. If the original optimization problem is a multi-objective one, the objectives are improved one at a time, converting the others to inequality constraints;
Step 4: The modified and improved population is now evaluated by the original fitness function to find the true score values. The scores return to the GeDEA algorithm, together with the locally improved population, replacing the old one;
Step 5: If the maximum number of generations is reached then stop, otherwise the GDMA algorithm performs the ranking through the GeDEA operands, creates the mating pool for the crossover and repeats the operations from the Step 3. The algorithm periodically retrain the global surrogate model, every $G_{NNtrain}$ generations, to improve and update the original fitness function approximation using the new available data.

4. Comparison with other multiobjective evolutionary algorithms

A systematic comparison of various MOEAs was originally provided in [70]. They compared the performance of eight algorithms (VEGA, HLGA, FFGA, NPGA, NSGA, SPEA, a random search algorithm and a single-objective EA using weighted-sum aggregation) on six test problems featuring the characteristics that may cause difficulties in converging to the Pareto optimal front and in maintaining diversity within the population [71]: convexity, non-convexity, discrete Pareto fronts, multimodality, deception and biased search spaces. The best overall results were obtained respectively with SPEA and NSGA. Further investigations demonstrated that an elitist variation of NSGA (the so-called NSGA-II) equals the performance of SPEA.

The original version of GeDEA algorithm [39] was compared following a similar procedure, showing that peculiar feature of preserving the diversity within the population can highly help to reach better results respect to both SPEA and NSGA. GeDEA demonstrated improved performance in terms of convergence and Pareto front coverage for all the functions considered.

In this section GDMA is tested on the same problems according to the same methodology. The results are compared using the same metric with the results coming from the original GeDEA and other state of the art evolutionary algorithms, SPEA2 [72] and NSGA-II [73], available for public use as part of the PISA framework at http://www.tik.ee.ethz.ch/sop/pisa. The test functions and the complete methodology will be briefly recall in the following chapters for clarity.

4.1. Test functions

The original methodology introduced by [70] involves six different functions, Tau1, . . ., Tau6, each one with a distinct feature, as identified by [71]. Each of those functions is a two-objective minimization problem constructed in the same way, according to the guidelines in [71]:

$$
\begin{aligned}
&\text{Minimize } \tau(x) = (f_1(x_1), f_2(\mathbf{x}))\\
&\text{subject to } f_2(\mathbf{x}) = g(x_2, \ldots, x_n)h(f_1(x_1), g(x_2, \ldots, x_n))\\
&\text{where} \quad \mathbf{x} = (x_1, \ldots, x_n)
\end{aligned}
\tag{2}
$$

The function g controls the search space lateral to the Pareto-optimal front, while the function f_1 controls the search space along the Pareto-optimal front. The function h determines the shape of the front itself. This method makes it possible to investigate the problem features separately in order to assess whether an MOEA has the ability to converge to the true Pareto-optimal set and to find diverse Pareto-optimal solutions under particular conditions. Since the deceptive problems are not supported by the GDMA algorithm (due to the presence of a gradient-based algorithm inside the optimization loop), the Tau5 function in [70] is not considered and quoted here.

- Test function Tau1 has a convex Pareto-optimal front:

$$
\begin{aligned}
&f_1(x_1) = x_1\\
&g(x_2, \ldots, x_n) = 1 + 9\sum_{i=2}^{n}\frac{x_i}{(n-1)}\\
&h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}
\end{aligned}
\tag{3}
$$

where $n = 30$ and $x_i \in [0,1]$. The Pareto-optimal front corresponds to $g(\mathbf{x}) = 1$.

- Test function Tau2 has a non-convex Pareto-optimal front:

$$
\begin{aligned}
&f_1(x_1) = x_1\\
&g(x_2, \ldots, x_n) = 1 + 9\sum_{i=2}^{n}\frac{x_i}{(n-1)}\\
&h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^2
\end{aligned}
\tag{4}
$$

where $n = 30$ and $x_i \in [0,1]$. The Pareto-optimal front corresponds to $g(\mathbf{x}) = 1$.

- Test function Tau3 has a discontinuous Pareto-optimal front consisting of several convex parts:

$$f_1(x_1) = x_1$$
$$g(x_2, \ldots, x_n) = 1 + 9\sum_{i=2}^{n} \frac{x_i}{(n-1)} \tag{5}$$
$$h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} - \left(\frac{f_1}{g}\right)\sin(10\pi f_1)$$

where $n = 30$ and $x_i \in [0,1]$. The Pareto-optimal front corresponds to $g(\mathbf{x}) = 1$.

- Test function Tau4 involves a multi-modal problem:

$$f_1(x_1) = x_1$$
$$g(x_2, \ldots, x_n) = 1 + 10(n-1) + \sum_{i=2}^{n}(x_i^2 - 10\cos(4\pi x_i)) \tag{6}$$
$$h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}$$

where $n = 10$, $x_1 \in [0,1]$ and $x_2, \ldots, x_n \in [-5,5]$. The Pareto-optimal front is convex and corresponds to $g(\mathbf{x}) = 1$.

- Test function Tau6 features non-uniformity of the search space:

$$f_1(x_1) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$$
$$g(x_2, \ldots, x_n) = 1 + 9\sum_{i=2}^{n} \frac{x_i}{(n-1)} \tag{7}$$
$$h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^2$$

where $n = 10$ and $x_i \in [0,1]$. The Pareto-optimal front is non-convex and corresponds to $g(\mathbf{x}) = 1$.

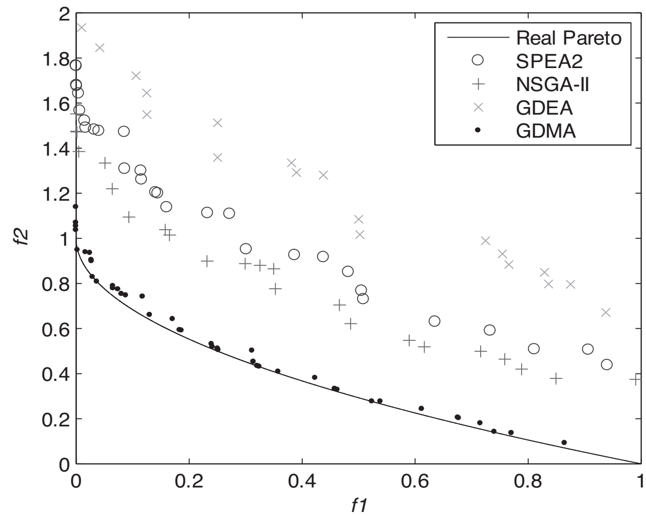
4.2. Metrics of performance

Several metrics can be considered to assess the performance of different EAs with respect to the different goals of optimization itself (refer to [70]): how far is the resulting non-dominated set from the true Pareto front, how uniform is the distribution of the solutions along the Pareto front, how wide is the Pareto front. Several metrics are used here. The first metric is a *Pareto dominance-based* technique, which is an assessment method based on pairwise comparisons of Pareto front sets, also known as Coverage of two sets or CTS metric [39,70]. The others are three *Quality indicators* [74] known as Generational Distance [75], Spacing Metric [73] and Maximum Pareto Front Error [75]. Following a brief explanation of the metrics used to assess the GDMA method.

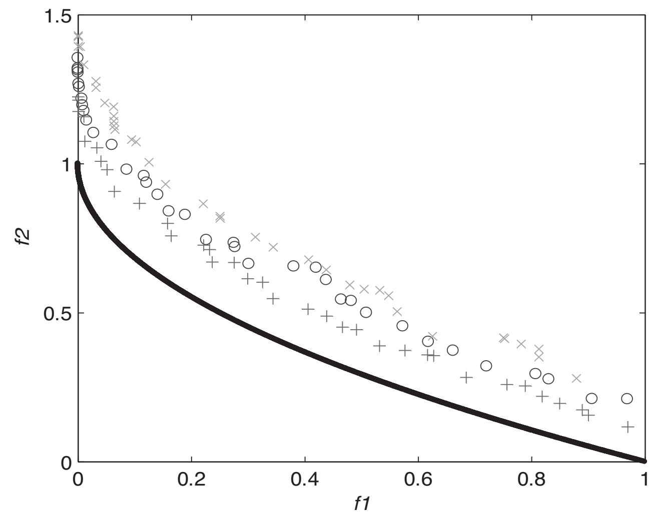
Coverage of two sets (CTS): Let $X', X'' \subseteq N$ be two sets of decision vectors. The function CTS maps the ordered pair (X', X'') to the interval [0,1]:

$$CTS(X', X'') = \frac{|\{\mathbf{x}'' \in X''; \exists \mathbf{x}' \in X' : F(\mathbf{x}') \prec F(\mathbf{x}'')\}|}{|X''|} \tag{8}$$

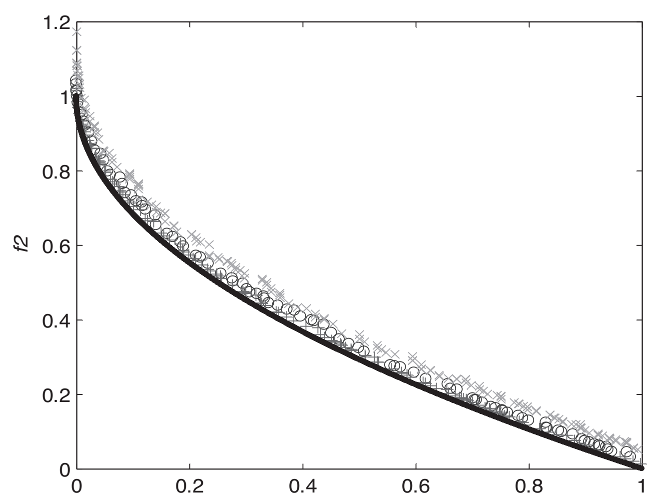
When $\text{CTS}(X', X'') = 1$ the set X'' is entirely covered by the solutions in X'. On the other hand, if $\text{CTS}(X', X'') = 0$, none of the solutions in X'' is covered by the set X'. The value of $\text{CTS}(X', X'')$ is not necessarily equal to $1 - \text{CTS}(X', X'')$ and both the possible ordered pairs are considered during the comparison. The selected metric, when applied to two non-dominated sets from different EAs, gives the percentage of solutions of the set X'' that are covered by solutions of the set X', although superiority is not estimated.



(a)



(b)



(c)

Fig. 3. Test function Tau1 (convex problem) after 20 (a), 40 (b) and 80 (c) generations.

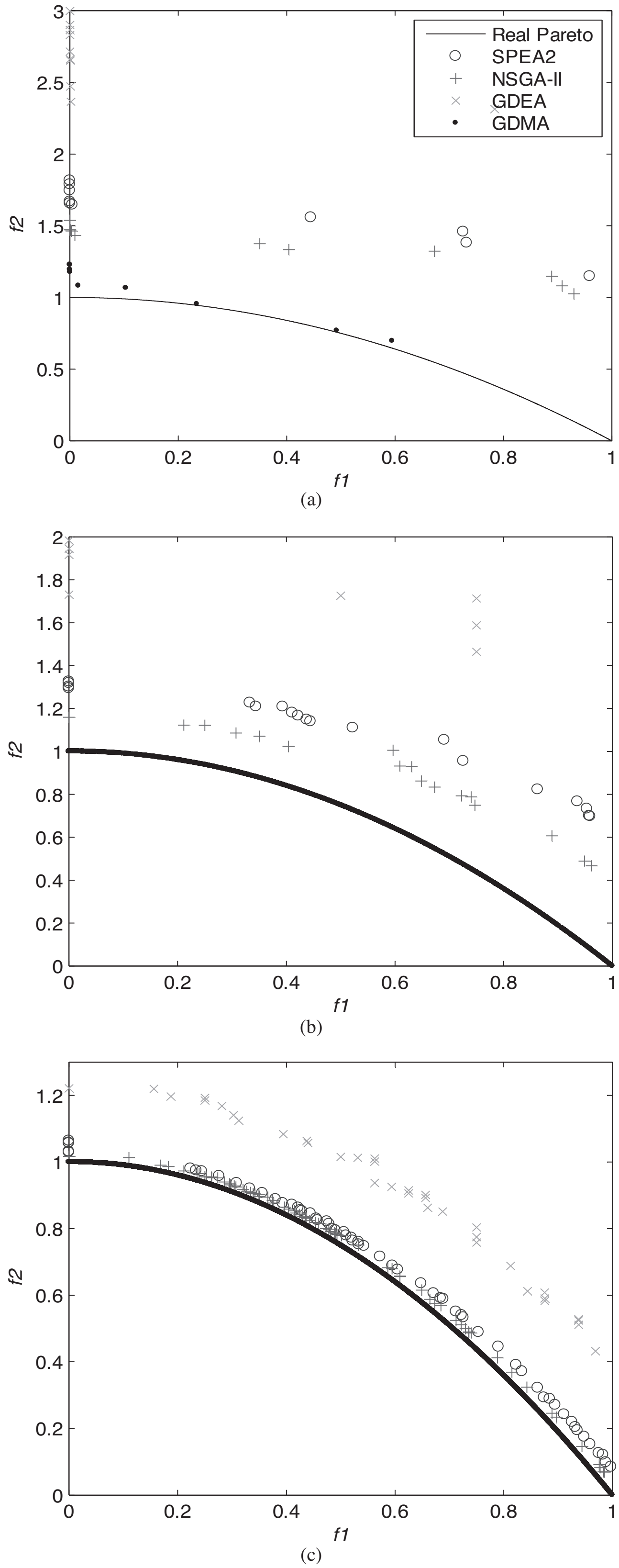


Fig. 4. Test function Tau2 (non-convex problem) after 20 (a), 40 (b) and 80 (c) generations.

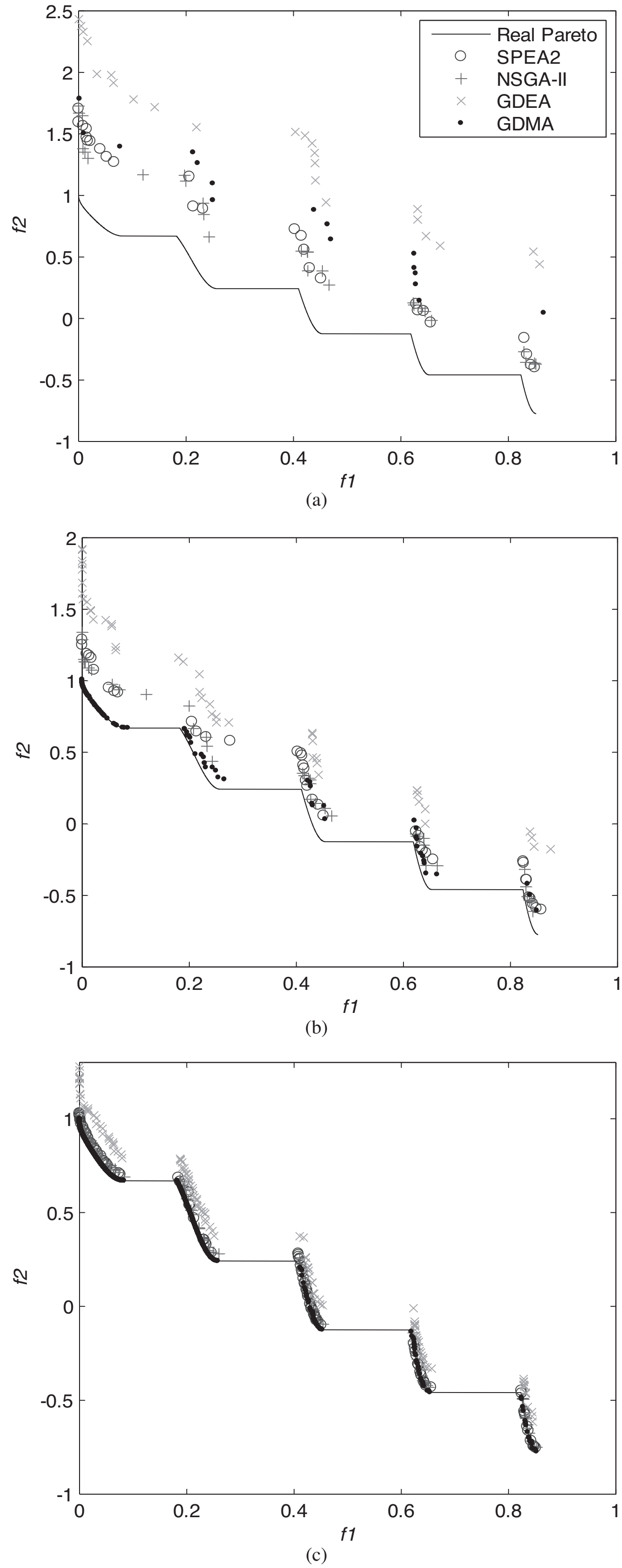


Fig. 5. Test function Tau3 (discrete problem) after 20 (a), 40 (b) and 80 (c) generations.

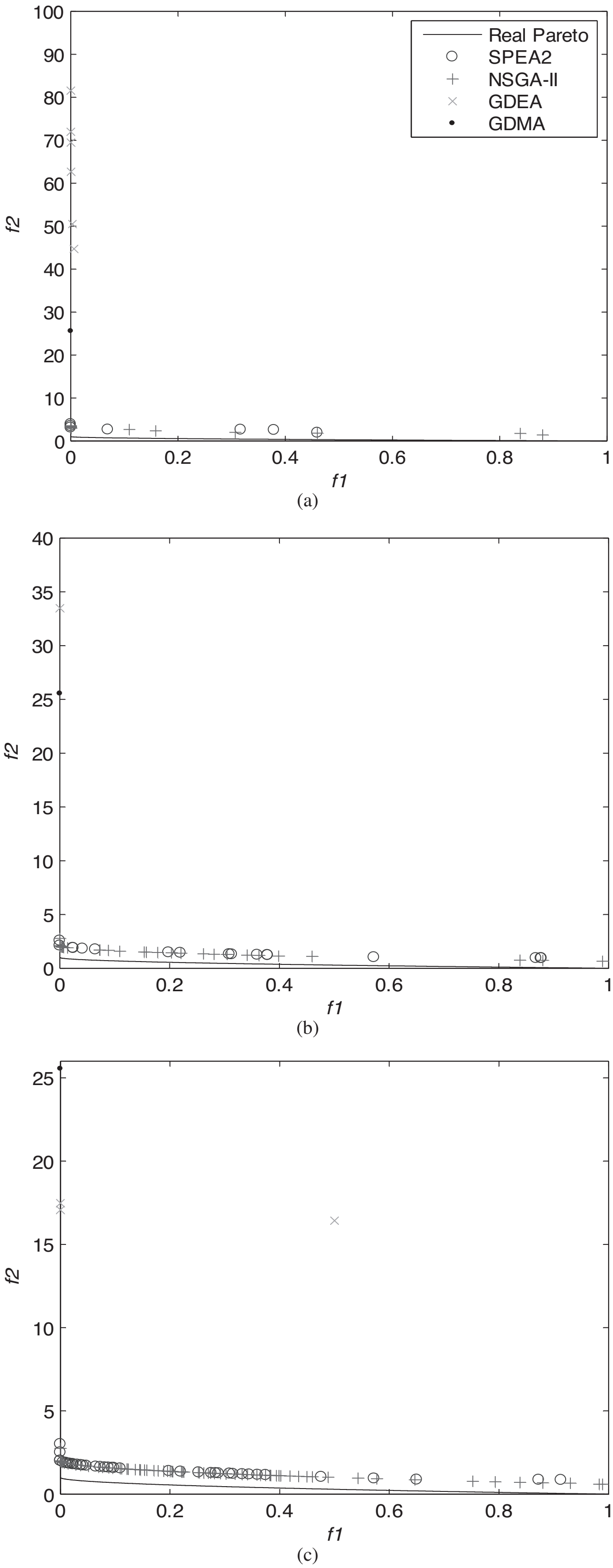


Fig. 6. Test function Tau4 (multi-modal problem) after 20 (a), 40 (b) and 80 (c) generations.

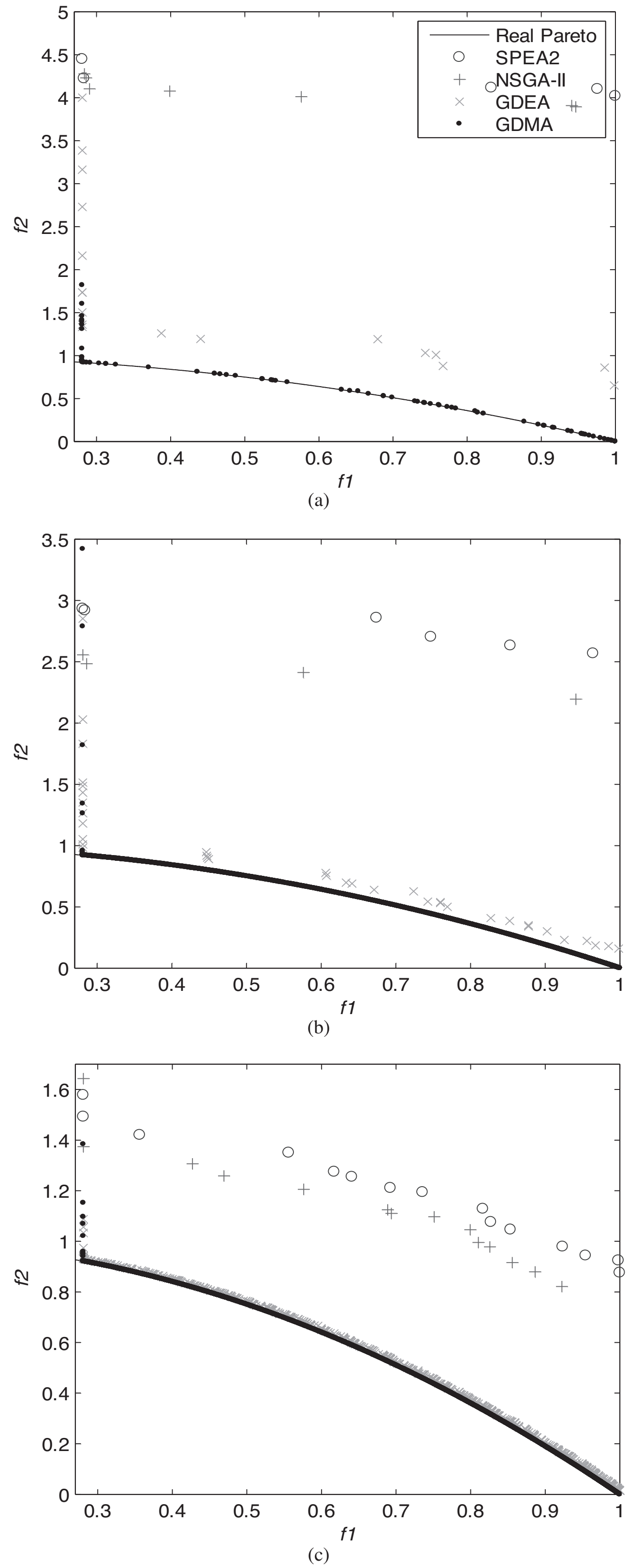


Fig. 7. Test function Tau6 (non-uniform problem) after 20 (a), 40 (b) and 80 (c) generations.

Generational Distance (GD): measures the root-mean-square of the distance of points in the approximation set from their nearest point in the true Pareto front. It is to be minimized.

Spacing Metric (SP): measures the average discrepancy between the spacing of consecutive points and the mean spacing of consecutive points. It is to be minimized.

Maximum Pareto Front Error (MPFE): measures the largest distance between a vector in the current Pareto front and the corresponding closest point in the true Pareto front. It is to be minimized.

The three quality indicators are calculated for all the 30 runs of each algorithm (see §4.3 for the methodology description) and statistical representations of them are plotted to compare the algorithms' performance.

4.3. Methodology

The methodology used in [39,70] is followed, with the additional feature of intermediate comparisons of the performance during the evolution of the computations. These further information are added to show and investigate the temporal convergence rate toward the real Pareto front of the algorithms under consideration. In this way the authors want to highlight not only the final convergence levels but also the rapidity demonstrated by GDMA reaching them. This is a topic feature when the optimization algorithm is strictly developed to match the demands for convergence rapidity, typical of practical engineering applications (e.g. aerodynamic shape optimizations using Computational Fluid Dynamics), that is the final aim of the present algorithm. On the present work GDMA is run 30 times on each test function and, in a similar manner, GeDEA, SPEA2 and NSGA-II. Every algorithm is started with 30 random populations, since the first population of SPEA2 and NSGA-II are not available as outcome. The non-dominated solutions among the populations generated during a run (offline performance) is considered as output for all the algorithms, accordingly to [70]. A total number of 80 generations is performed, but the offline Pareto front is also calculated after 20 and 40 generations respectively. The original number of generations used in [39,70] was 250, but several reasons led to choose a lower value equal to 80: first of all the increasing performance of the algorithms lead to a nearly complete convergence after such a big number of generations, with several Pareto front sets almost coincident or visibly indistinguishable. Secondly, GDMA is specifically intended to engineering applications where the execution of 250 generations is most of the time unfeasible due to the computational time effort. The population size μ is set to 100 and the size of offspring population λ is set to 100 as a $\mu + \mu$ evolution strategy. For SPEA2 and NSGA-II the crossover probability is set to 100% for coherence. The mutation probability p_{mut} is set to 0.01. All the algorithms work with a uniform binary crossover and each parameter is represented by 30 bits. All the six functions in [70] are analyzed except the Tau5 function (deceptive problem) which is not supported by GDMA.

The GDMA requires several additional parameters. One is the starting GeDEA generations, G_{GeDEA}, equal to10. It is the number of initial generation performed only with the global searcher (or

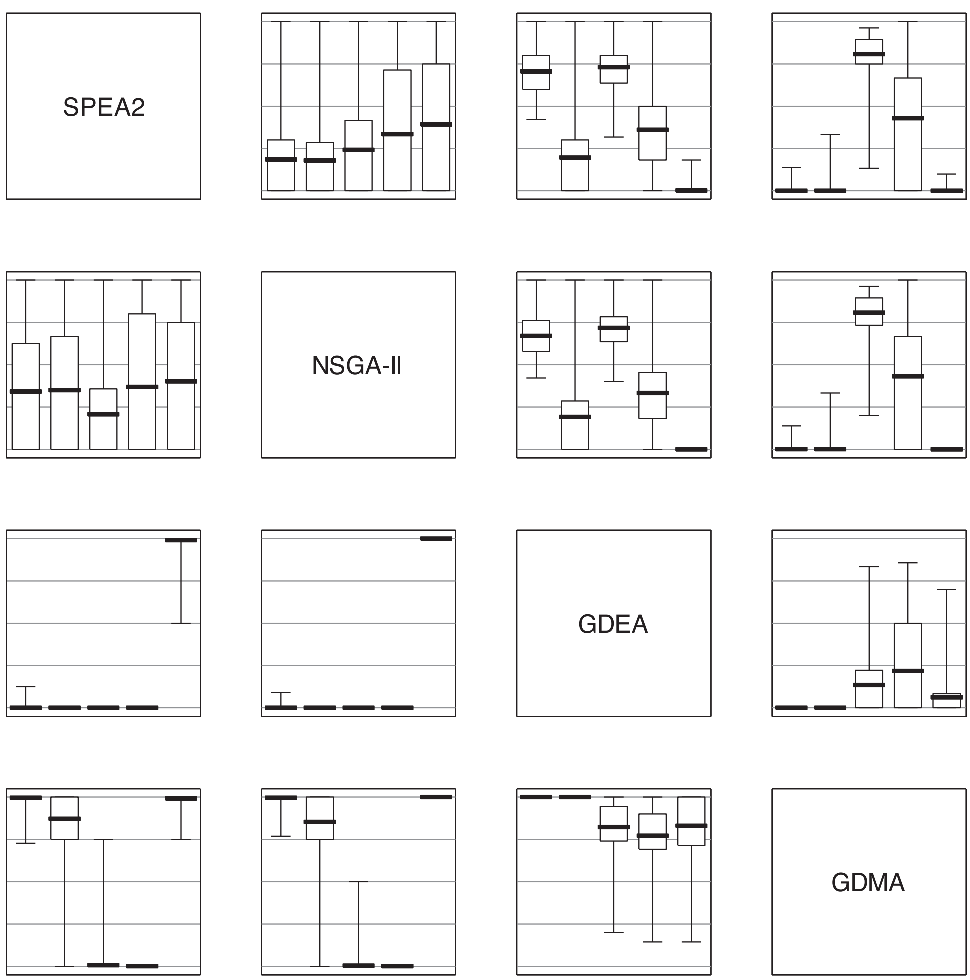


Fig. 8. Box plots based on the CTS metric after 20 generations. Each square contains five box plots representing the distribution of CTS values for the five test functions, from Tau1 (leftmost) to Tau6 (rightmost). The scale is 0 at the bottom and 1 at the top in all squares.

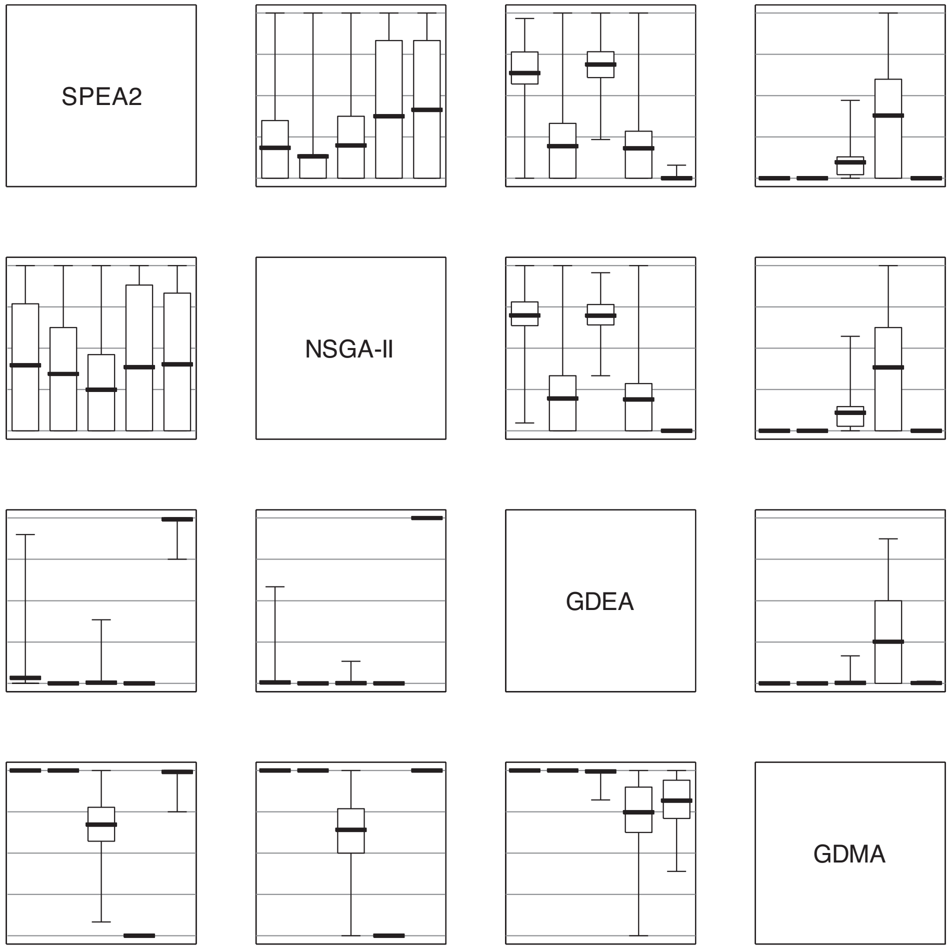


Fig. 9. Box plots based on the CTS metric after 40 generations.

GA). They are necessary and GDMA cannot be started immediately because it requires a sufficient database size for the ANN training. Second is the number of GDMA generations between two ANN trainings, $G_{NNtrain}$, which is set to10.

5. Experimental results and discussions

GDMA demonstrates similar performance for both Tau1 (convex) and Tau2 (non-convex) problems, respectively in Figs. 3 and 4. In general, for non-convex functions the algorithm requires slightly more generations to approach and reach a good coverage of the true Pareto front. This is probably due to the technique adopted during the gradient-based refinement. In both the cases the convergence is complete already after 40 generations and the following calculations lead just to an improvement of the Pareto density. The final non-dominated solutions cover completely the true Pareto front with a very small distance between two close solutions. The percentage of the Pareto set solutions respect to the total number of evaluations after 80 generations is about 53% and 46% respectively for convex and non-convex problems. Looking at the box plots it is clear that the other three algorithms included in the comparison never dominate the Pareto front found by GDMA, neither partially, since the metric CTS(X,GDMA) is almost always equal to 0. Observing the remaining metrics for such kind of problems (Figs. 11 and 12) the GDMA's excellent performance are confirmed since the algorithm was able to reach null value (best value possible) for the three quality indicators already after 40 generations with better results also after 20.

The response of GDMA for Tau3 (discrete problem) is slightly worse than what obtained for Tau1 and Tau2 but in general good. In this case the metric CTS(X,GDMA) decreases gradually with the generations up to zero at the end of the computations. During the first part of the optimizations the ANN (due to the reduced number of individuals available within the database) has some difficulty to accurately approximate the original fitness function. The metric CTS(GDMA,X) in this case is always lower than 1 and decreases gradually with the number of evaluations. The solutions in the final non-dominated set are around 17% (average) of the total evaluations. The coverage of the true Pareto front (visible in Fig. 5 for different numbers of generations) is complete but the density is not equally distributed on it, indeed the most of the solutions are localized for low values of the first objective, while at higher values, above 0.5, the distance between adjacent non-dominated solutions increases. GDMA always provides better performance (based on GD, SP and MPFE) than the other algorithms for the Tau3 just after 40 generations.

GDMA demonstrated inability to treat multi-modal problems (Tau4 visible in Fig. 6). This is due to the insufficient accuracy of the surrogate model representing the original function. Such a problem is confirmed also by the three quality indicators in Fig. 14 that provides larger values respect to the other algorithms.

The performance with Tau6 (non-uniform problem) are the best obtained in the present work and they are visible in Fig. 7. The explanations are the excellent approximation obtained with the surrogate model and also the already good performance of GeDEA with this kind of function. Already after 20 generations the

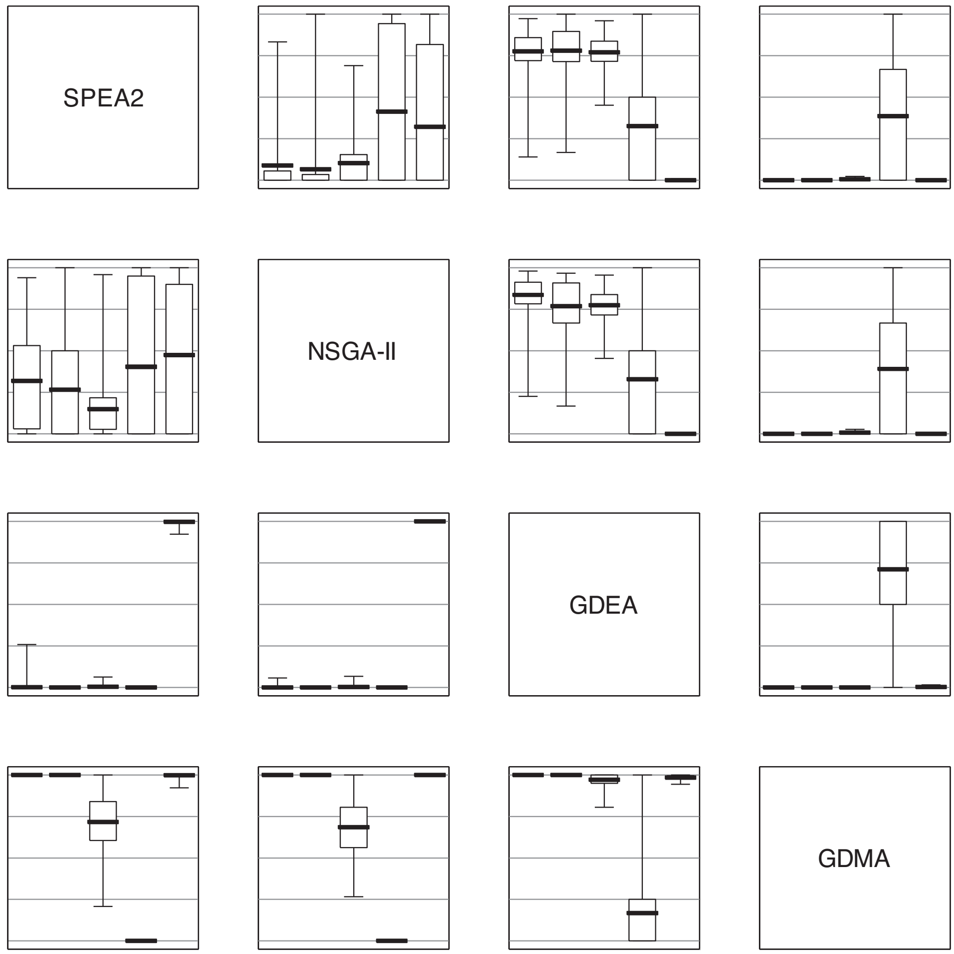


Fig. 10. Box plots based on the CTS metric after 80 generations.

non-dominated set is very close to the true Pareto front and the number of solutions that lie on it increase gradually up to about 60% of the total database at the end of one computation. CTS(X,GDMA) is always equal to 0 and conversely the CTS(GDMA,X) is almost always equal to 1, with few exceptions. Fig. 15 shows the algorithms' performance with Tau6 based on GD, SP and MPFE metrics. GD and SP are almost always 0 after 40 generations for GDMA while the other algorithms remains higher. The MPFE metric looks larger for GDMA. This is due to the Pareto front solutions at minimum $f1$ (see Fig. 7 for a qualitative explanation) which often have high value of the $f2$, representing a large distance from the real Pareto curve (Figs. 8–15).

6. GDMA application to helicopter airfoil design

The proposed algorithm is also tested on a real engineering problem in order to prove its superiority on practical problems and highlight its advantages over concurrent methods. The problem consists in the aerodynamic performance optimization of an airfoil for helicopter rotor application. In order to perform such an optimization the GDMA algorithm is coupled to the Fluent fluid-dynamic solver and a structured mesher to generate 2D airfoil grids. The airfoil's surface is parameterized using a b-spline approach that allows the complete description of an airfoil using 14 control points. The baseline airfoil SC1095 (UH-60A main rotor airfoil) is selected as a reference point for the optimizations to be carried out. The parameterization method applied to the SC1095 is shown in Fig. 16 (left) together with the related mesh (right). It

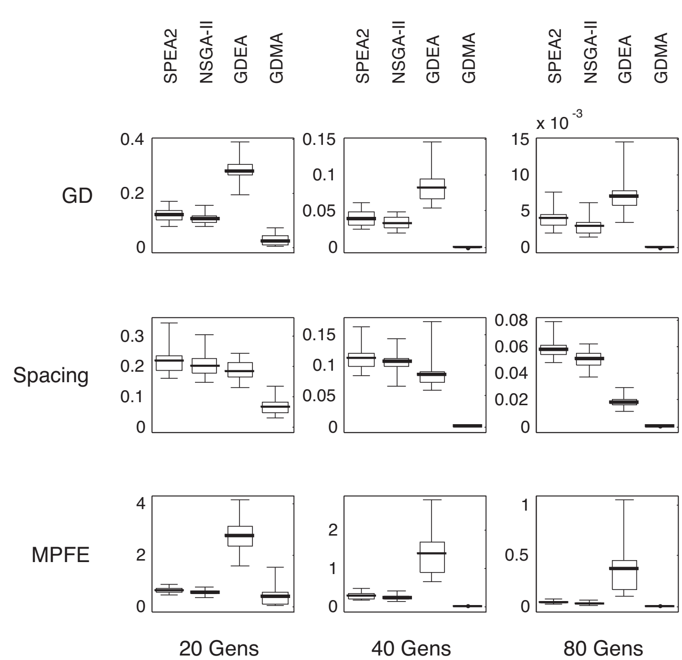


Fig. 11. GD, SP and MPFE metrics after 20, 40 and 80 generations for test function Tau1.

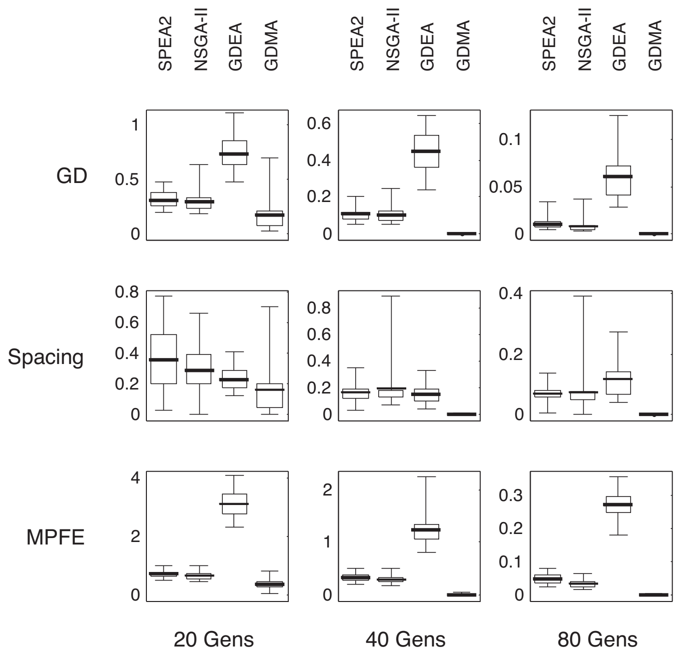


Fig. 12. GD, SP and MPFE metrics after 20, 40 and 80 generations for test function Tau2.

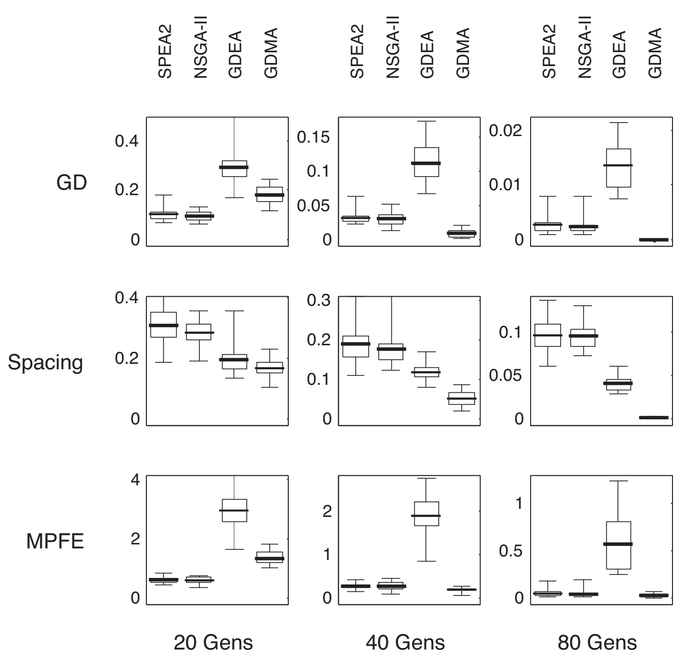


Fig. 13. GD, SP and MPFE metrics after 20, 40 and 80 generations for test function Tau3.

is a 500 × 150 structured mesh with O-grid topology. The boundary layer flowfield is fully resolved (y^+ always below 1) using the one-equation Spalart–Allmaras turbulence model. The flow is considered fully turbulent, thus no laminar-turbulent boundary layer transition simulated.

The problem under consideration is a multi-point/multi-objective one having two objectives (with several constraints) related to two different airfoil operative condition. Eq. (9) describes the objective function to be minimized. Each term of the vector is composed of a main objective, which is the minimization of the aerodynamic drag coefficient (for two separated conditions, refer to Table 1), and several inequality constraints handled using the

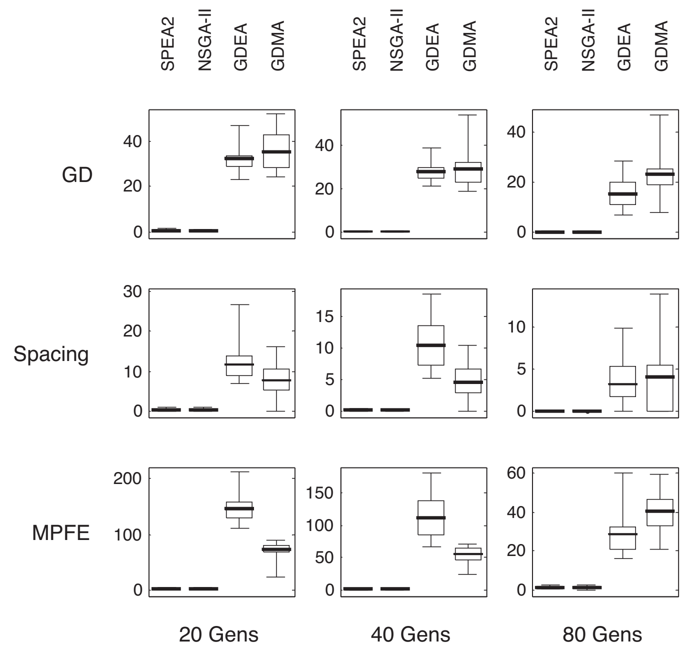


Fig. 14. GD, SP and MPFE metrics after 20, 40 and 80 generations for test function Tau4.

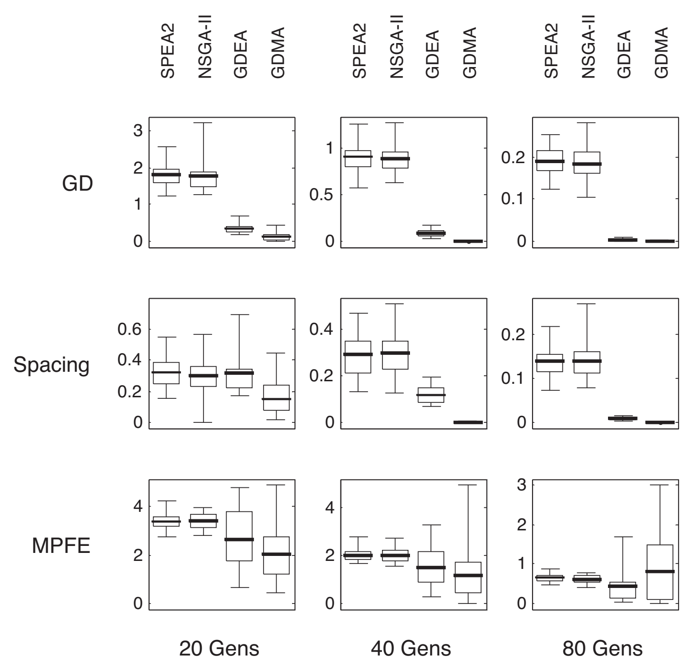


Fig. 15. GD, SP and MPFE metrics after 20, 40 and 80 generations for test function Tau6.

Table 1
Freestream conditions for airfoil optimization.

Condition	M	α [°]
1	0.6	5
2	0.4	9

penalty function approach described in Eq. (10). In the equation p represents the variable to be constrained and p^* is the reference value. β and q are parameters of the power function which have been set respectively 1 and 0.85 for all the problem's constraints under consideration. The first objective of the MOOP contains an

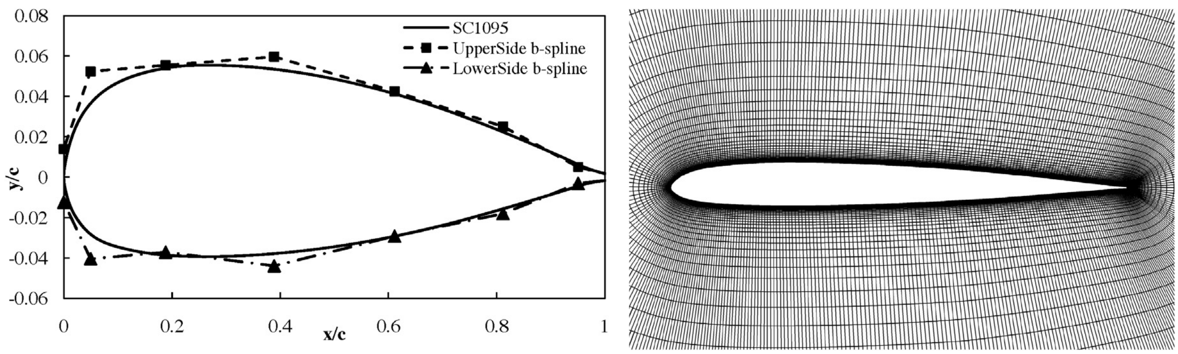


Fig. 16. Airfoil parameterization using b-splines (left) and structured mesh used during the optimizations (right).

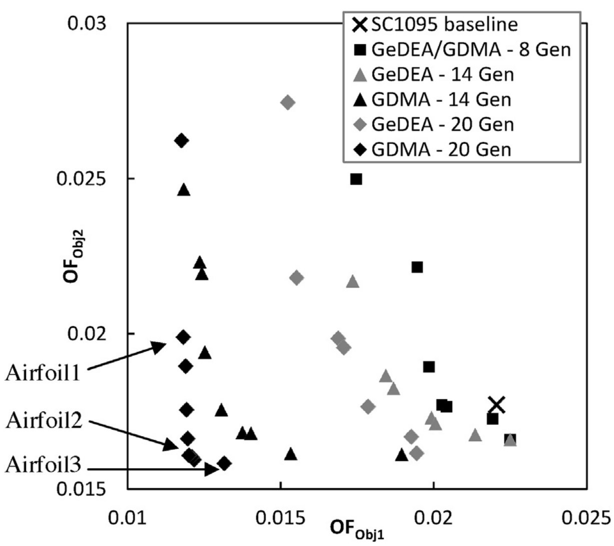


Fig. 17. Outcomes of the GDMA and GeDEA comparison on the 2D airfoil optimization problem.

aerodynamic constraint ensuring that the lift coeff. for condition 1 is equal to or higher than the SC1095 lift, and a geometrical constraint ensuring that the optimal airfoils always have a thickness higher than 9.5% (SC1095). The second objective only contains the aerodynamic constraints in lift coeff. related to the condition 2 and a similar constraint in lift with respect to objective 1.

$$OF = \begin{Bmatrix} C_{d,Cond1} + PF_{C_{l,Cond1}} + PF_{tc} \\ C_{d,Cond2} + PF_{C_{l,Cond2}} \end{Bmatrix} \tag{9}$$

$$PF = \begin{cases} 0 & \text{if} \quad p \geq p^* \\ \beta |p - p^*|^q & \text{otherwise} \end{cases}, \tag{10}$$

GDMA is applied to the aforementioned engineering problem and compared to a more common GA, which is GeDEA. In order to conduct a fair comparison the same number of function calls are used, equal to 800, corresponding to 20 generations with 40 individuals each. The assessment is conducted by qualitatively comparing the offline Pareto fronts after 8, 14 and 20. Since GDMA requires an initial database of individuals to train the surrogate model (ANN), the first 8 generations by GeDEA are used as starting point. The offline Pareto fronts are plotted in Fig. 17 together with the baseline performance of the SC1095 airfoil. The GeDEA Pareto fronts are plotted in gray using triangles and diamonds for 14 and 20 generations respectively. The GDMA are plotted in a similar fashion in black. It is worth noting that GDMA produced a remarkable advancement in objective space already after 14 generations while

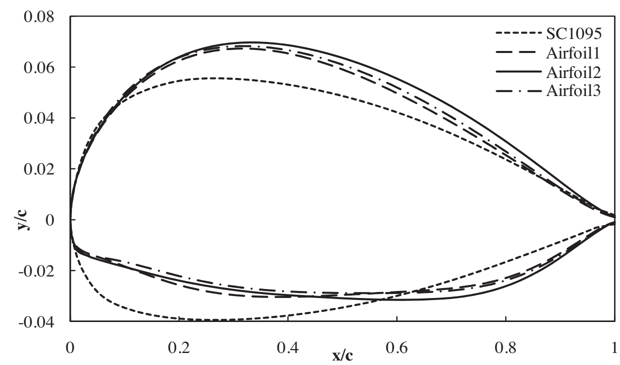


Fig. 18. Examples of optimal airfoils selected among the final Pareto (GDMA optimization).

GeDEA was not able to reach the same improvements after 20 generations, thus meaning improved efficiency and excellent capacity in global optimization. The results show what was clear also in the previous assessment of the algorithm, which is the great rapidity in convergence toward the real Pareto. In addition the algorithm seems to work well also in presence of Pareto fronts having a sharp corner, which are in general more complex than others concerning the Pareto front coverage and spacing.

Some of the optimal airfoils coming from the GDMA final Pareto front are also plotted in Fig. 18 compared to the baseline SC1095.

7. Conclusion

A new surrogate-assisted memetic algorithm has been proposed in the present work. Its main novel characteristic is the handling of the diversity as an objective during the evolutionary process. This algorithm has been extensively tested and compared with other evolutionary algorithms on commonly used benchmark functions, to demonstrate its ability solving multi-objective problems. The outcomes from the experimental tests lead to the following conclusions:

- GDMA outperforms the other state-of-the-art evolutionary algorithms considered in the comparison for the following types of problem: convex, non-convex, discrete and non-uniform. The best advantages are visible during the first 20 generations, in which the convergence level is improved with respect to the other algorithms.
- For convex, non-convex, discrete and non-uniform functions, GDMA demonstrated to be able to reach a complete convergence to the true Pareto-optimal front already after 40 generations, when the other algorithms are usually still far away from it. This ability makes it suitable for usage with high computationally

expensive fitness functions and in all the cases in which the efficiency of the optimizer plays an essential role.

- The Pareto-optimal front coverage and the distribution uniformity of the non-dominated solutions are surprisingly good. GDMA covers entirely, in most of the cases, the true Pareto-optimal front with uniform distance (in the objective space) between adjacent solutions belonging to the optimal set. This fact is mainly due to the large percentage of non-dominated solutions respect to the total number of fitness function evaluations, leading to very dense optimal sets. After 80 generations the optimal set can contain from about 20% up to 60% of the total amount of solutions.
- The repeatability of the final solution is a very important aspect on the assessment of a new evolutionary algorithm, since its nature is intrinsically connected to randomness. The final optimal set found by GDMA is never covered by any other algorithm (for convex, non-convex, discrete and non-uniform functions). The statistical analysis of the data suggests that the algorithm normally converges up to the true Pareto-optimal front, reaching a complete convergence.
- The multi-modal problem is one case in which GDMA demonstrated to be unable to reach good convergence level and to be inferior than the other algorithms. Further investigations have identified the surrogate model as main problem, since it is not able to approximate with sufficient accuracy multi-modal functions. The problem could be solved in future works introducing multi-surrogate strategies with automatic adaptation to the function under consideration.
- GDMA has been tested successfully on a two-point/two-objective problem with constraints involving an airfoil's aerodynamic optimization. The algorithm was tested against the GeDEA algorithm only, showing also for practical optimization problems a clear superiority in reaching optimal solution faster and deeper.

References

[1] C.A. Coello Coello, G.B. Lamont, D.A. van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems Series: Genetic and Evolutionary Computation, 2nd ed., 2007, ISBN:978-0-387-33254-3.
[2] Y. Jin, M. Olhofer, B. Sendhoff, A framework for evolutionary optimization with approximate fitness function, IEEE Trans. Evol. Comput. 6 (5) (2002) 481–494.
[3] Y.S. Ong, P.B. Nair, A.J. Keane, Evolutionary optimization of computationally expensive problems via surrogate modelling, Am. Inst. Aeronaut. Astronaut. J. 41 (4) (2003) 687–696.
[4] A.I.J. Forrester, A.J. Keanea, Recent advances in surrogate-based optimization, Prog. Aerosp. Sci. 45 (January–April (1–3)) (2009) 50–79.
[5] R.H. Myers, D.C. Montgomery, Response Surface Methodology, 2nd ed., Wiley-Interscience, New York, 2002.
[6] G.E. Box, N.R. Draper, Empirical Model-Building and Response Surfaces, Wiley, New York, 1987.
[7] M.F. Hussain, R.R. Barton, S.B. Joshi, Metamodeling: radial basis functions, versus polynomials, Eur. J. Oper. Res. 138 (1) (2002) 142–154.
[8] R.L. Eubank, Nonparametric Regression and Spline Smoothing, Marcel Dekker, Inc., New York, 1999.
[9] A.W. Bowman, A. Azzalini, Applied Smoothing Techniques for Data Analysis, Clarendon Press, Oxford, UK, 1997.
[10] P.J. Green, B.W. Silverman, Nonparametric Regression and Generalized Linear Models, Chapmal & Hall, 2–6 Boundary Row, London, UK, 1994.
[11] J.H. Friedman, Multivariate adaptive regression splines, Ann. Stat. 19 (1) (1991) 1–67.
[12] S.C. Raisinghani, A.K. Ghosh, P.K. Kalra, Two new techniques for parameter estimation using neural networks, Aeronaut. J. 102 (1011) (1998) 25–29.
[13] A.A. Mullur, A. Messac, Extended radial basis functions: more flexible and effective metamodeling, AIAA J. 43 (6) (2005) 1306–1315.
[14] C. Praveen, R. Duvigneau, Radial Basis Functions and Kriging Metamodels for Aerodynamic Optimization, INRIA Report No. 6151, 2007.
[15] A.A. Mullur, A. Messac, Extended radial basis function: more flexible and effective metamodeling, AIAA J. 43 (6) (2005) 1306–1315.
[16] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, 2000.
[17] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Stat. Comput. 14 (3) (2004) 199–222.
[18] J.A.K. Suykens, G. Horvath, S. Basu, C. Micchelli, J. Vandewalle (Eds.), Advances in Learning Theory: Methods, Models and Applications, NATO Science Series III: Computer & Systems Sciences, vol. 190, IOS Press Amsterdam, 2003.
[19] N.A.C. Cressie, The origins of kriging, Math. Geol. 22 (1990) 239–252.
[20] I. Clark, W.V. Harper, Practical Geostatistics, Ecosse North America, Columbus, Ohio, 2000.
[21] A.I.J. Forrester, A.J. Keane, N.W. Bressloff, Design and analysis of 'noisy' computer experiments, AIAA J. 44 (10) (2006) 2331–2339.
[22] J. Hu, Methods of generating surfaces in environmental GIS applications, in: Proceedings of 1995 ESRI Conference, Palm Springs, California, USA, 1995, Available online from: http://proceedings.esri.com/library/userconf/proc95/to100/p089.html
[23] V.V. Toropov, U. Schramm, A. Sahai, R.D. Jones, T. Zeguer, Design optimization and stochastic analysis based on the moving least squares method, in: 6th World Congress of Structural and Multidisciplinary Optimization, Rio de Janeiro, 30th May–3rd June, 2005.
[24] W.S. Cleveland, Robust locally weighted regression and smoothing scatter plots, J. Am. Stat. Assoc. 74 (1979) 829–836.
[25] J. Fan, T. Gasser, I. Gijbels, M. Brockmann, J. Engel, Local polynomial regression: optimal kernels and asymptotic minimax efficiency, Ann. Inst. Stat. Math. 49 (1) (1997) 79–99.
[26] J. Fan, I. Gijbels, Local Polynomial Modelling and Its Applications, Chapman & Hall, 1996.
[27] D. Ruppert, S.J. Sheather, M.P. Wand, An effective bandwidth selector for local least squares regression, J. Am. Stat. Assoc. 90 (432) (1995) 1257–1270.
[28] R. Smith, B. Dike, S. Stegmann, Fitness inheritance in genetic algorithms, in: ACM Symposiums on Applied Computing, 1995, pp. 345–350.
[29] J.H. Chen, D.E. Goldberg, S.Y. Ho, K. Sastry, Fitness inheritance in multi-objective optimization, in: Genetic and Evolutionary Computation Conference, 2002, pp. 319–326.
[30] H.S. Kim, S.B. Cho, An efficient genetic algorithms with less fitness evaluation by clustering, in: Congress on Evolutionary Computation, 2001, pp. 887–894.
[31] Y. Jin, B. Sendhoff, Reducing fitness evaluations using clustering techniques and neural networks ensembles, in: Genetic and Evolutionary Computation Conference, LNCS 3102, 2004, pp. 688–699.
[32] P.K.S. Nain, K. Deb, Computationally effective search and optimization procedure using coarse to fine approximation, in: Congress on Evolutionary Computation, 2003, pp. 2081–2088.
[33] G. Chi-Keong, T. Kay Chen, Multi-Objective Memetic Algorithms Springer Series on Studies in Computational Intelligence, vol. 171, Springer, Berlin/Heidelberg, 2009, ISBN:978-3-540-88050-9.
[34] N. Krasnogor, H. Ishibuchi, IEEE Trans. Syst. Man Cybern. – Part B: Spec. Issue Memet. Algorithms 37 (February (1)) (2007).
[35] D. Lim, J. Yaochu, O. Yew-Soon, B. Sendhoff, Generalizing surrogate-assisted evolutionary computation, IEEE Trans. Evol. Comput. 14 (3) (2010) 329–355, http://dx.doi.org/10.1109/TEVC.2009.2027359
[36] D. Lim, J. Yaochu, O. Yew-Soon, B. Sendhoff, A study on metamodeling techniques, ensembles and multi-surrogates in evolutionary computation, in: GECCO'07, 2007, pp. 1288–1295, http://dx.doi.org/10.1145/1276958.1277203
[37] Y.S. Ong, P.B. Nair, K.Y. Lum, Hybrid evolutionary algorithm with hermite radial basis function interpolations for computationally expensive adjoint solvers, Comput. Optim. Appl. 39 (1) (2008) 91–119, http://dx.doi.org/10.1007/s10589-007-9065-5
[38] K.C. Giannakoglou, D.I. Papadimitriou, I.C. Kampolis, Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels, Comput. Methods Appl. Mech. Eng. 195 (2006) 6312–6329, http://dx.doi.org/10.1016/j.cma.2005.12.008
[39] A. Toffolo, E. Benini, Genetic diversity as an objective in multi-objective evolutionary algorithms, Evol. Comput. 11 (2) (2003) 151–167.
[40] T. Friedrich, P.S. Oliveto, D. Sudholt, C. Witt, Analysis of diversity-preserving mechanisms for global exploration, Evol. Comput. 17 (4) (2009) 455–476.
[41] M. Lozano, F. Herrera, J.R. Cano, Replacement strategies to preserve useful diversity in steady-state genetic algorithms, Inf. Sci. 178 (23) (2008) 4421–4433, http://dx.doi.org/10.1016/j.ins.2008.07.031
[42] J. Botzheim, A novel diversity induction method for bacterial memetic algorithm by hibernation of individuals, in: 6th International Conference on Genetic and Evolutionary Computing, 2012, pp. 328–331, http://dx.doi.org/10.1109/ICGEC.2012.25
[43] K. Sörensen, M. Sevaux, MA|PM: memetic algorithms with population management, Comput. Oper. Res. 33 (5) (2006) 1214–1225, http://dx.doi.org/10.1016/j.cor.2004.09.011
[44] K.A. De Jong, An Analysis of the Behavior of a Class of Genetic Adaptive Systems (Ph.D. dissertation), Univ. of Michigan, Ann Arbor, 1975.
[45] S.W. Mahfoud, Niching Methods for Genetic Algorithms (Ph.D. dissertation), University of Illinois, Urbana-Champaign, 1995.
[46] O.J. Mengshoel, D.E. Goldberg, Probabilistic crowding: deterministic crowding with probabilistic replacement, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99), 1999, pp. 409–416.
[47] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. 21 (6) (1953) 1087–1092.
[48] G. Harik, Finding multimodal solutions using restricted tournament selection, in: 6th Int. Conf. Genetic Algorithms, 1995, pp. 24–31.
[49] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[50] O.J. Mengshoel, D.E. Goldberg, The crowding approach to niching in genetic algorithms, Evol. Comput. 16 (3) (2008) 315–354.

[51] S.W. Mahfoud, Niching methods, in: Handbook of Evolutionary Computation, 1997, C6.1:1–C6.1:4.
[52] D.E. Goldberg, J. Richardson, Genetic algorithms with sharing for multimodal function optimization, in: Proceedings of the Second International Conference on Genetic algorithms and Their Application, 1987, pp. 41–49.
[53] P. Snijders, E.D. De Jong, B. de Boer, F. Weissing, Multi-objective diversity maintenance, in: GECCO'06, 2006.
[54] L.T. Bui, J. Branke, H.A. Abbass, Diversity as a selection pressure in dynamic environments, in: GECCO'05, 2005.
[55] E. Collingwood, D. Corne, P. Ross, Useful diversity via multiploidy, in: Proceedings of IEEE International Conference on Evolutionary Computation, 1996, pp. 810–813, http://dx.doi.org/10.1109/ICEC.1996.542705
[56] H. Shimodaira, DCGA: a diversity control oriented genetic algorithm, in: Second International Conference On Genetic Algorithms in Engineering Systems: Innovations and Applications, GALESIA 97, 1997, pp. 444–449, http://dx.doi.org/10.1049/cp:19971221
[57] I.K. Evans, Enhancing recombination with the complementary surrogate genetic algorithm, in: Proceedings of the IEEE International Conference on Evolutionary computation, 1997, pp. 97–102, http://dx.doi.org/10.1109/ICEC.1997.592276
[58] C.K. Ting, H.K. Buning, A mating strategy for multi-parent genetic algorithms by integrating tabu search, in: Proceedings of the 2003 Congress on Evolutionary Computation, 2003, pp. 1259–1266, http://dx.doi.org/10.1109/CEC.2003.1299813
[59] M. Hutter, Fitness uniform selection to preserve genetic diversity, in: Proceedings of the 2002 Congress on Evolutionary Computation, 2002, pp. 783–788, http://dx.doi.org/10.1109/CEC.2002.1007025
[60] J. Tang, M.H. Lim, Y.S. Ong, Diversity-adaptive parallel memetic algorithm for solving large scale combinatorial optimization problems, Soft Comput. 11 (9) (2007) 873–888, http://dx.doi.org/10.1007/s00500-006-0139-6
[61] J. Tang, M.H. Lim, Y.S. Ong, Adaptation for parallel memetic algorithm based on population entropy, in: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, GECCO'07, 2007, pp. 575–582, http://dx.doi.org/10.1145/1143997.1144100
[62] A.V. Kononova, K.J. Hughes, M. Pourkashanian, D.B. Ingham, Fitness diversity based adaptive memetic algorithm for solving inverse problems of chemical kinetics, in: IEEE Congress on Evolutionary Computation, CEC 2007, 2007, pp. 2366–2373, http://dx.doi.org/10.1109/CEC.2007.4424767
[63] K. Sörensen, M. Sevauxb, MA|PM: memeticalgorithms with populationmanagement,", Comput. Oper. Res. 33 (5) (2006) 1214–1225, http://dx.doi.org/10.1016/j.cor.2004.09.011
[65] M.Y.M. Ahmed, N. Qin, Surrogate-based aerodynamic design optimization: use of surrogates in aerodynamic design optimization, in: 13th International Conference on Aerospace Sciences & Aviation Technology, ASAT-13, May 26–28, 2009.
[66] M.T. Hagan, H.B. Demuth, M. Beale, Neural Network Design, PWS Publishing Company, 1996.
[67] M.D. Schmid, A Neural Network Package for Octave – User's Guide - Version: 0.1.9.1, 2009.
[68] R. Fletcher, M.J.D. Powell, A rapidly convergent descent method for minimization, Comput. J. 6 (1963) 163–168.
[69] D. Goldfarb, A family of variable metric updates derived by variational means, Math. Comput. 24 (1970) 23–26.
[70] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evol. Comput. 8 (2) (2000) 173–195.
[71] K. Deb, Multi-objective genetic algorithms: Problem difficulties and construction of test problems, Evol. Comput. 7 (3) (1999) 205–230.
[72] E. Zitzler, M. Laumanns, L. Thiele, Spea2: Improving the Strength Pareto Evolutionary Algorithm, Technical Report 103, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, 2001.
[73] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002).
[74] E. Zitzler, L. Thiele, J.D. Knowles, A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers, TIK-report No. 214, Computer Engineering and Networks Laboratory, ETH Zurich, 2006.
[75] D.A. Van Veldhuizen, Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations (Ph. D. thesis), Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, 1999.