2015 IEEE Symposium Series on Computational Intelligence

Comparative Study of Recent Multimodal Evolutionary Algorithms

Romaric Pighetti
Univ. Nice Sophia Antipolis,
CNRS, I3S, UMR 7271,
06900 Sophia Antipolis, France
Email: pighetti@i3s.unice.fr

Denis Pallez
Univ. Nice Sophia Antipolis,
CNRS, I3S, UMR 7271,
06900 Sophia Antipolis, France
Email: denis.pallez@unice.fr

Frédéric Precioso
Univ. Nice Sophia Antipolis,
CNRS, I3S, UMR 7271,
06900 Sophia Antipolis, France
Email: precioso@i3s.unice.fr

***Abstract*—Multimodal Optimization (MMO) aims at identifying several best solutions to a problem whereas classical optimization converge often to only one good solution. MMO has been an active research area in the past years and several new evolutionary algorithms have been developed to tackle multimodal problems.**

In this work, we compare extensively three recent evolutionary algorithms (MoBiDE, Multimodal NSGAII and MOMMOP). Each algorithm uses multiobjectivization, together with niching techniques to address single objective MMO problems. We have fully re-implemented MoBiDE and MM-NSGAII in order to better evaluate their sensitivity to parameter changes and their strengths and weaknesses. We have carefully evaluated all algorithms on the same benchmark functions and with the same parameters settings. The algorithms are also compared to a non-multimodal evolutionary algorithm to better highlight the impact of the multimodal adaptations.

Moreover, full access to the detailed results and source code is granted on our website for the ease of reproducibility.

I. Introduction

MultiModal Optimization Problems (MMOP) are problems for which several optimal solutions exist. Identifying different optimal solutions allows the user to choose which one fits his needs the best. Indeed, engineers often face problems in which physical conditions, prices or time can make one or several solutions unreachable. In those situations, having a set of optimal or near optimal solutions allows to switch to what fits the best depending on the situation. Therefore, the need of finding several optimal solutions is becoming more and more present. However, classical evolutionary optimization methods usually focus on finding only one solution. Therefore, the development of multimodal optimization (MMO) algorithms has been an intense research area in the past years.

Several techniques have been developed to tackle this problem [1], [2], [3], [4]. The study of all those techniques is out of the scope of this work, which focus on the comparative study of 3 recent evolutionary algorithms dedicated to multimodal optimization : NSGAII adapted for MMO (PNA-NSGAII) [5] , Multimodal Optimization using a Bi-objective Differential Evolution (MOBiDE) [6] and Multiobjective Optimization for MultiModal Optimization Problems (MOMMOP) [7]. To solve a multimodal problem, these algorithms transform the multimodal scalar problem into a multiobjective optimization problem (MOP). This process is referred to as *multiobjectivization*. They also use niching techniques, which consist in finding and preserving several solutions to the problem in different places of the search space. To stress the importance of the multimodal adaptations, we also compare the studied algorithms to a classic multiobjective genetic algorithm. We chose the original version of NSGAII as the multiobjective genetic algorithm, and it is used to solve the same bi-objective problem as PNA-NSGAII.

To assess the performances of such algorithms, several benchmark functions have been built [8]. In spite of this profusion of benchmark functions, experimental conditions may differ from one study to another. So, even though the number of function evaluations is often the same, a lot of other parameters come into play for each algorithm performances. Thus, comparing results of different algorithms with slightly different parameters could lead to wrong interpretations of the results. Moreover, most of the time only the best performances obtained are presented. No information is given about the sensitivity to parameters, which can be very important for real cases.

All comparisons are done under the strict same experimental conditions and statistical tests were conducted to validate them. In addition, the influence of a specific parameter of MOBiDE is explored to better understand the strengths and weaknesses of this algorithm. As no implementation of PNA-NSGAII and MOBiDE were available, we implemented our own version of those algorithms. The implementations have been made in java using ECJ [9]. All results and material used for the experimentations are available on our website[1].

The next section introduces the tested algorithms. Experimental conditions are presented in details in Section III. Section IV presents and discusses the results obtained. Finally, Section V summarizes and concludes the paper.

II. Algorithms

In this section, we will briefly introduce each of the compared algorithms and their workflow after introducing some concepts about multiobjective problems. For the full details about each algorithm, the reader can refer to the corresponding papers: NSGAII [10], PNA-NSGAII [5], MOBIDE [6] and MOMMOP [7].

[1]http://denispallez.i3s.unice.fr/doku.php?id=soft_multimodal

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.124

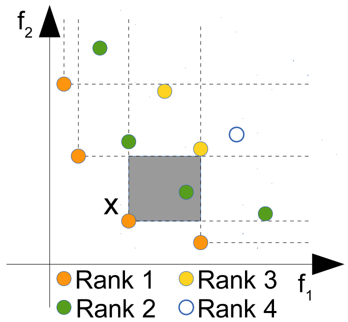

Fig. 1: Example of non dominated sorting in a minimization context. The gray colored rectangle represents the hypervolume measure of the individual $\mathbf{X}$

A. Multiobjective Problem

Each studied algorithm uses multiobjectivization to solve MMOPs. The result of this operation is a Multiobjective Optimization Problem (MOP) which solutions are solutions to the MMOP.

The formalization of a MOP is shown in Def. 1. Solving a MOP consists in finding parameter vectors that optimize (maximize or minimize depending on the context) F.

Definition 1. *Multiobjective optimization problem*

Let $F = [f_1, f_2, ..., f_n]$,
$\mathcal{D} = \{x \in \mathbb{R}^m : h(x) = 0, g(x) \geq 0\}$.
$minF(x), x \in \mathcal{D}$ *is a multi-objective optimization problem. Elements of* $\mathcal{D}$ *are called parameter vectors, potential solutions (sometimes shorten as solutions) or individuals (in the context of evolutionary algorithms).*

In order to evaluate the performance of parameter vectors to solve a MOP, the dominance (more precisely pareto dominance) relation has been created and is explained in Def. 2. If a parameter vector is dominated by another parameter vector, it is a poorer solution to the MOP. It is worth noting that sometimes we cannot compare two potential solutions using the dominance relation. It is illustrated in Fig.1 in the case of a 2 objectives problem where all parameter vectors having the same color do not dominate each other.

Definition 2. *Pareto dominance*

Let $F = \{f_1, f_2, ..., f_n\}$ *be a n objectives problem from* $\mathcal{D}$ *in* $\mathbb{R}^n$ *to be minimized. For* $(\mathbf{x_1}, \mathbf{x_2}) \in \mathcal{D}$, $\mathbf{x_1}$ *dominates* $\mathbf{x_2}$, *noted* $\mathbf{x_1} \prec \mathbf{x_2}$, *if and only if:*
$\forall i \in [1, n]$, $f_i(\mathbf{x_1}) \leq f_i(\mathbf{x_2})$ *and*
$\exists i \in [1, n]$ / $f_i(\mathbf{x_1}) < f_i(\mathbf{x_2})$

Using the dominance relation, the non-dominated sorting procedure consists in assigning a rank to each element of a set of parameter vectors. This procedure is presented in Alg. 1. Fig.1 shows an example of the result of this procedure, each color representing a rank.

The pareto front is then defined as the subset of individuals of rank 1 within a set of individuals. The optimal pareto front is the set of elements from $\mathcal{D}$ that are non dominated. This is the solution to a MOP. Thus this is what a multiobjective algorithm should converge to. We will now explain the key components of the studied algorithms.

Algorithm 1 Non dominated sorting of a set of parameter vectors S

```
rank = 1
while S ≠ ∅ do
    H = nonDominatedIndividual(S)
    assign rank to H
    rank = rank + 1
    S = S \ H
end while
```

B. PNA-NSGAII

The goal of PNA-NSGAII is to optimize a scalar multimodal function, i.e. finding different parameter vectors that give the optimum value of the function. To do so, multiobjectivization is used. A second objective, ensuring diversity, is added to the function to be optimized. This objective is presented in (1), where POP is the size of the population. $\Omega(\mathbf{x})$ measures the cumulative distance to other individuals in the population, so, the smaller f_2 is, the further the individual is from other individuals. Therefore, minimizing f_2 increases the diversity in the population.

$$f_2(\mathbf{x}) = \frac{1}{\sum_{j=1}^{POP} ||\mathbf{x} - \mathbf{x_j}||^2} = \frac{1}{\Omega(\mathbf{x})} \quad (1)$$

To solve this bi-objectives problem (the MMOP under study, referred to as f, and the introduced f_2), a dedicated multiobjective algorithm is used. For PNA-NSGAII, a modified version of NSGAII is used. The workflow of PNA-NSGAII is presented in Figure 3. A niching technique has been added in the non-dominated sorting routine of NSGAII. It consists in making individuals that are far from one another not comparable. This leads to a new definition of the dominance relation given in Def. 3.

Definition 3. *PNA-NSGAII dominance*

For $(\mathbf{x_1}, \mathbf{x_2}) \in \mathcal{D}$, $\mathbf{x_1}$ *dominates* $\mathbf{x_2}$, *noted* $\mathbf{x_1} \prec_{mm} \mathbf{x_2}$, *if and only if:*
$proximate(\mathbf{x_1}, \mathbf{x_2})$ *is true and*
$\forall i \in [1, n]$, $f_i(\mathbf{x_1}) \leq f_i(\mathbf{x_2})$ *and*
$\exists i \in [1, n]$ / $f_i(\mathbf{x_1}) < f_i(\mathbf{x_2})$

This prevents the algorithm from discarding individuals that are far from any other individual in the population (because they are part of the pareto front with the new dominance relation), and thus maintain a good diversity of individuals through the search space. The proximate function computation is given by (2), (3) and (4). m is the dimension of the problem, U_d and L_d are the upper and lower bounds of the search space for dimension d and $\mathbf{x_d}$ stands for the d$^{\text{th}}$ component of vector $\mathbf{x}$.

$$T = \exp\frac{ln(POP)}{m} \quad (2)$$

$$v_d = \frac{U_d - L_d}{T} \forall d \in 1, 2, ..., m \quad (3)$$

$$proximate(\mathbf{x_1}, \mathbf{x_2}) = 1 \text{ if } |(\mathbf{x_1} - \mathbf{x_2})_d| \leq v_d \quad (4)$$

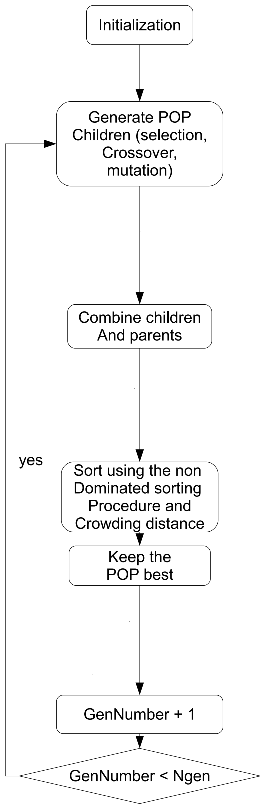

Fig. 2: NSGAII workflow

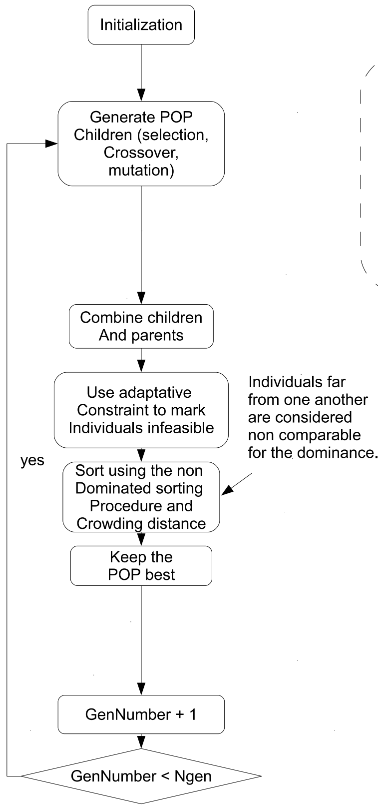

Fig. 3: PNA-NSGAII workflow

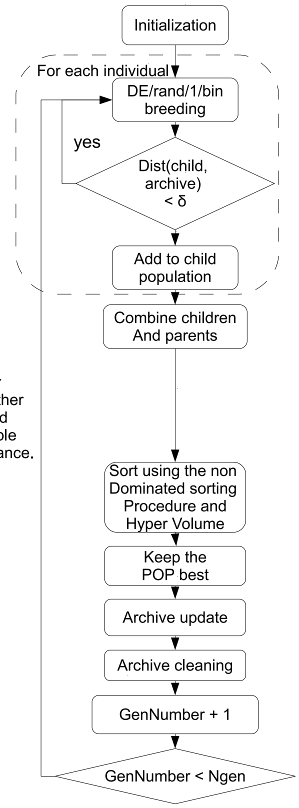

Fig. 4: MOBiDE's workflow

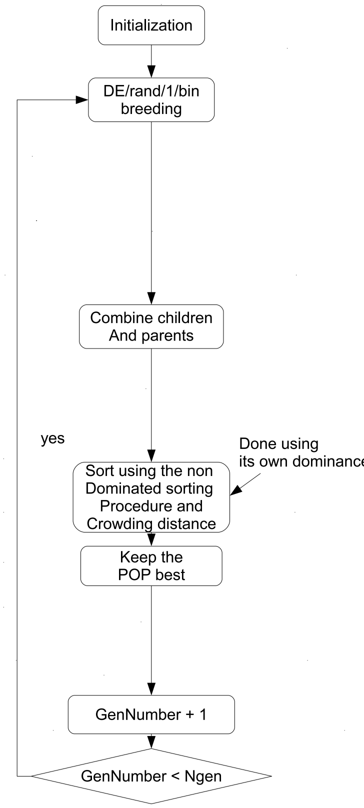

Fig. 5: MOMMOP workflow

TABLE I: Multi-objectivization of the different algorithms

	NSGAII	PNA-NSGAII	MOBiDE	MOMMOP
Objective 1	Min. or Max. $(f(x))$			Min. $\left(x_i + \frac{\lvert f(\mathbf{x}) - F^{best}\rvert}{\lvert F^{worst} - F^{best}\rvert} \times (U_i - L_i) \times \eta\right)$
Objective 2	Min. $\left(f_2(\mathbf{x}) = \frac{1}{\sum_{j=1}^{POP} \lvert\lvert \mathbf{x} - \mathbf{x_j} \rvert\rvert^2} = \frac{1}{\Omega(\mathbf{x})}\right)$		Max. $\left(f_3(\mathbf{x_d}) = \frac{\Omega(\mathbf{x})}{POP} = \frac{1}{POP \times f_2(\mathbf{x})}\right)$	Min. $\left(1 - x_i + \frac{\lvert f(\mathbf{x}) - F^{best}\rvert}{\lvert F^{worst} - F^{best}\rvert} \times (U_i - L_i) \times \eta\right)$

Preventing the comparison of individuals far from one another allows groups of individuals to gather around local optima. To prevent this and guide the individuals toward the global optima, another self-adaptive parameter is introduced, the adaptive constraint. It is used to force the convergence toward the best values by making individuals with a fitness worse than a computed threshold infeasible. Equations (5) to (7) show how this threshold is adaptively computed through the generations of the GA. (5) is presented in a minimization context. ϵ is the expected precision for the retrieved optima, F^{best} is the best value of $f(\mathbf{x})$ found so far and $MaxGen$ is the maximum number of generation allowed for the evolution process. Looking at f_{gen} that exponentially decreases with the generations, we can see that this constraint is loose at the beginning of the run and tightens as the end of the run approaches. This makes the population progressively gather and converge toward the global optima as local optima become infeasible.

$$\text{If } f(\mathbf{x}) > F^{best} + f_{gen} \times \epsilon \text{ then } \mathbf{x} \text{ is infeasible.} \tag{5}$$

$$f_{gen} = a \times \exp^{b \times gen} \text{ with:} \tag{6}$$

$$a = \frac{10^{14}}{\exp^b},\ b = \frac{\ln(2) - \ln(10^{14})}{MaxGen} \tag{7}$$

Now that PNA-NSGAII has been presented, we will describe the principal components of MOBiDE.

C. MOBiDE

As for PNA-NSGAII, MOBiDE introduces a second objective focusing on the diversity of solutions. This objective is presented in (8). We can see that this objective is almost the same as the one added in PNA-NSGAII.

$$\begin{aligned} &\text{Max.} \left(f_3(\mathbf{x_d}) = \frac{\Omega(\mathbf{x})}{POP}\right) \\ \iff &\text{Max.} \left(\frac{1}{POP \times f_2(\mathbf{x})}\right) \end{aligned} \tag{8}$$

What differs between those algorithms is the evolutionary algorithm used (DE for MOBiDE and a GA for PNA-NSGAII) and the niching techniques. MOBiDE's workflow is presented in Fig.4.

MOBiDE is based on DE/rand/1/bin [11] to generate new individuals. The breeding procedure of this algorithm is recalled in Alg. 2. MOBiDE uses an external archive to keep track of the best individuals found through the evolution. Each time an individual is generated, if it is within a δ radius of any element of the archive, it is immediately discarded, and another individual is generated. This makes the population concentrate on different part of the search space, which is a kind of niching technique. Then, individuals of the population

Algorithm 2 DE Breeding procedure

for $i \in [1, POP]$ **do**
  chose i_1, i_2, i_3 randomly such that $i \neq i_1 \neq i_2 \neq i_3$
  $V = X_{i_1} + F \times (X_{i_2} - X_{i_3})$
  $U = binaryCrossover(V, X_i)$
  keep best of U and X_i for the next generation.
end for

are ranked using the non dominated sorting procedure and the hypervolume criteria. The hypervolume is a measure of diversity in the objective space. It consists in measuring the difference of the hypervolume dominated by the current rank and the hypervolume dominated by the current rank without the studied solution. This gives the volume of space dominated only by the studied solution in the rank. Fig.1 illustrates this in a 2 objectives minimization case. The gray rectangle represents the hypervolume measure of solution $\mathbf{x}$. The greater the hypervolume measure is, the better the solution. The hypervolume measure is used to sort individuals belonging to the same rank. After the ranking, only the POP best individuals are kept to form the new population.

The archive is then updated using this new population. Individuals that present a good enough fitness for the multimodal function (9) are added to the archive. The required fitness value to be added to the archive is computed from the best fitness value found since the beginning of the run (F^{best}) and a fixed parameter α. α has been set to 0.1 in the experiments, as it is the case in the work presented in [6]

Add $\mathbf{x}$ to the archive if

$$\begin{cases} f(\mathbf{x}) < (1+\alpha) * F^{best} \text{ when } F^{best} > 0 \\ f(\mathbf{x}) < (1-\alpha) * F^{best} \text{ when } F^{best} < 0 \\ f(\mathbf{x}) < 0.001 \text{ when } F^{best} \approx 0 \end{cases} \tag{9}$$

Finally, the archive is cleansed by removing any individual that has a fitness bellow the average fitness of the individuals in the archive. Then a new generation can begin with the generation of new individuals.

D. MOMMOP

PNA-NSGAII and MOBiDE add an objective to the existing multimodal function to be optimized to build a Multiobjective Optimization Problem (MOP). MOMMOP adopts another strategy. It builds a brand new MOP, using the multimodal problem as part of the different objectives. The Bi-Objective Problem (BOP) built by MOMMOP is presented in (10). In each BOP, η is a parameter that grows with the number of function evaluations (FEs), giving more importance to the optimization of the MMOP f as the run progresses.

$$BOP_i \begin{cases} \text{Min.} \left(x_i + \frac{|f(\mathbf{x}) - F^{best}|}{|F^{worst} - F^{best}|} \times (U_i - L_i) \times \eta \right) \\ \text{Min.} \left(1 - x_i + \frac{|f(\mathbf{x}) - F^{best}|}{|F^{worst} - F^{best}|} \times (U_i - L_i) \times \eta \right) \end{cases} \tag{10}$$

Authors of MOMMOP proved that the two objectives of this MOP are conflicting. Indeed, the first objective scales with x_i while the second scales with $1 - x_i$. And for both objectives, the second term is the same and is a normalized evaluation of f. So these two objectives are clearly conflicting. Authors of MOMMOP also proved that all pareto optimal solutions to this MOP are solutions to the MMOP f. This means that the solutions to BOP_i is a subset of the solutions to f.

We can see that the presented MOP is associated with one of the decision variable. Choosing only one particular variable and solving the MOP associated with it could result in missing some of the solutions to the problem f. Indeed, several solutions to f may have the same value for x_i, and thus will have the same value for both objectives of BOP_i. Thus some solutions are more likely to be discarded by the algorithm (when ranking using the crowding distance and selecting only the best individuals). Therefore, MOMMOP is using simultaneously the bi-objective problems linked to each decision variable. To do so, an extension to the dominance relation has been defined and replaces the standard dominance in this algorithm. (11) shows this new dominance relation.

$$\begin{aligned} &\mathbf{x} \prec \mathbf{y} \text{ if :} \\ &\forall i \in [1, D], \mathbf{x} \prec \mathbf{y} \text{ on } BOP_i \end{aligned} \tag{11}$$

Then a niching technique, showed in (12) has also been added, modifying further the dominance. The normalization is explained in (13) and the distance is the euclidean distance. This modification helps discarding individuals that are too close to one another by making one dominates the other. So the dominated individual will be assigned a lower rank during the non dominated sorting, and thus is most likely to be discarded.

$$\begin{aligned} &\mathbf{x} \prec \mathbf{y} \text{ if :} \\ &f(\mathbf{x}) \text{ is better than } f(\mathbf{y}) \wedge \\ &distance(norm.(\mathbf{x}), norm.(\mathbf{y})) < 0.01 \end{aligned} \tag{12}$$

$$norm.(\mathbf{x}) = \mathbf{y} \text{ so that}$$

$$\forall i \in [1, D], y_i = \frac{x_i - L_i}{U_i - L_i} \tag{13}$$

The workflow of the algorithm used to solve the crafted MOP using this new dominance is presented in Fig.5. We can see that this algorithm is almost the same as NSGA-II, but instead of using genetic operators to generate new individuals, it uses the differential evolution operators. In addition, when sorting the population, the dominance described in this subsection is used instead of the standard dominance.

E. NSGAII

As explained before, all studied algorithms dedicated to MMOP use multiobjectivization. But then dedicated algorithms are created to solve the resulting multiobjective problem. To stress what is brought by those algorithms, we compared them to the original NSGAII [10], solving the MOP resulting from PNA-NSGAII's multiobjectivization.

NSGAII workflow is presented in Fig.2. In NSGAII, selection and generation of individuals is done using classic operators: a tournament selection, SBX crossover [12] and polynomial mutation [13].

Once the children are generated, the non dominated sorting procedure takes place on the combined population composed of both the parents and the children. Then, only the POP best individuals are kept to form the population for the next generation. The crowding distance is used to rank individuals belonging to a same rank. The less crowded an individual is, i.e. the bigger the crowding distance is and the better it is. The crowding distance computation procedure is recalled in Alg. 3. Finally, a new generation can take place with the generation of new individuals from the new population.

Algorithm 3 Compute the crowding distance for a front $\mathcal{F}$

```
l = |F|
for all i do
    set F[i]_crowding = 0
end for
for all objective m do
    F = Sort(F, m)
    F[1]_crowding = F[l]_crowding = ∞
end for
for i = 2 to l − 1 do
    F[i]_crowding = F[i]_crowding + (F[i+1].m − F[i−1].m) / (f_m^max − f_m^min)
end for
```

The components of each of the compared algorithms have been presented. Tab. I sum up the objectives used by each algorithm while the global workflow of each algorithm is presented in Figs. 2 to 5. We can now introduce the details about the experiments conducted.

III. EXPERIMENTS

This section aims at presenting the experimental conditions used. We will first describe common elements to all tested algorithms before introducing some specific parameters for each algorithm.

A. Common Elements

The first thing to set up is the set of benchmark functions that will be used. A large choice of functions is available [8]. To have a good insight of each algorithm strengths and weaknesses, it is essential to use functions that present different types of problems. A set of benchmark functions respecting this condition has been published for the CEC2013 multimodal optimization competition [14]. Along with the benchmark functions, this technical report also provides the maximum number of function evaluations allowed for each function, some interesting figures to present as results when dealing with MMOP and some experimental recommendations. We will use this benchmark as the test bed for our comparisons.

As recommended in [14], each experiment is run 50 times and average results are presented. We want to compare the average results of the same experiment done with several algorithms. To ensure that the differences observed between those means are relevant, a statistical test is required. We chose to use the Kruskal-Wallis [15] test to validate our results. This test is the equivalent to the ANOVA test when one does not know if the results of the experiments follow a normal law. In addition, the Kruskal-Wallis test presents similar results to the ANOVA test when dealing with data following a normal law and having enough samples [15].

TABLE II: Common parameters

Benchmark functions	CEC2013 [14]
Number of runs	50
Population size	$100 * dim$
Precision (ϵ)	0.001

To assess the performances of the algorithms, the peak ratio measure is used. This measure, presented in [14], is the ratio of the number of optima retrieved over the total number of optima of the function.

All implemented algorithms are population based. To be fair in our comparisons, we chose to compare them with the same population size. This size has been set to $100 * m$ where m is the dimension of the problem. This value is what has been used by the authors of [5].

In all the conducted experiments, the expected precision for the retrieved optima is $\epsilon = 0.001$.

Thanks to our own implementation of PNA-NSGAII and MOBiDE using the ECJ library, the matlab sources of MOMMOP provided by the authors and the implementation of NSGAII provided in ECJ, we have access to the results of each algorithm at each generation of the runs. So, in addition to the final results, we are also able to compare the results at each generation, and thus able to compare the convergence speed and convergence scheme of each algorithm. To ensure that those comparisons are relevant, the Kruskal-Wallis test is performed for each generation.

All the common elements have been presented, they are briefly summed up in Tab. II. We will now introduce some specific parameters of each algorithm under study and the values that were used for them.

B. Algorithm Specific Parameters

In this subsection, we present specific parameters inherent to each algorithm. We will also provide some precision about the decisions we made when implementing those algorithms. Indeed, some details were not fully provided and thus we had to take decisions about the expected behavior of the algorithms in those cases.

1) PNA-NSGAII: This algorithm adopts a self-adaptive niching technique, using the constraints on the search space (the size of the search space) and the size of the population. No new parameter is introduced besides the classic parameters of genetic algorithms such as mutation and crossover probability for example. All of these parameters are provided by the authors [5] and will be set without any changes in the remaining of this paper. They are recalled in Tab. III.

TABLE III: PNA-NSGAII specific parameters

SBX crossover probability (p_c)	0.9
Polynomial mutation probability (p_m)	0.05
SBX distribution index (ν_c)	10
Mutation distribution index (ν_m)	50

TABLE IV: MOBiDE specific parameters

F (p_c)	0.8
binomial crossover probability (Cr)	0.9
α (9)	0.1

2) MOBiDE: We saw before that MOBiDE introduces a parameter δ for its niching technique in addition to the standard parameters. Authors stated that this parameter must be small ($< 1e^{-3}$) without providing much more precision. We investigate the influence of this parameter by setting it to several values, $0.9e^{-3}$, $0.9e^{-4}$, $0.9e^{-5}$ and $0.9e^{-6}$.

In the third case of (9), no precision is given to check the equality to 0. We decided that the test will be done with a precision of $1e^{-3}$. Since any individual with a fitness below $1e^{-3}$ is added to the archive in such a case, it seems good to consider that a fitness below $1e^{-3}$ is considered close enough to 0 to apply the rule.

All other parameters were provided by the authors [6] and are used for all the experiments. They are summed up in Tab. IV.

3) MOMMOP: All the parameters of MOMMOP were specified by the authors and were used as is. They are recalled in Tab. V

IV. Results and discussions

Two main experiments have been conducted in this work. The first one aims at comparing the performances of PNA-NSGAII [5], MOBiDE [6] and MOMMOP [7], with NSGAII [10] used as a baseline to show the improvements brought by the multimodal adaptations. The second experiment aims at showing the impact of δ in MOBiDE. Tab. VI to IX compare the evolution of the peak ratio obtained by MOBiDE, PNA-NSGAII, MOMMOP and NSGAII for 4 functions from the CEC2013 benchmark [14]. The results of the Kruskal-Wallis tests for this experiment are reported in Tab. X, listing for each function, the generation numbers for which no conclusion can be drawn from the presented data. That is the confidence in the results was less than 95%. Tab. XI to XIII present the evolution of the peak ratio of MOBiDE with different δ values for 3 functions from the CEC2013 benchmark [14]. As for the previous experiment, the Kruskal-Wallis tests results are reported in Tab. XIV. The results of only 3 functions are shown here since other functions didn't bring more informations.

TABLE V: MOMMOP specific parameters

F (p_c)	0.5
Cr	0.7
η (10)	$40 \times D \times (CurrentFEs/MaxFEs)$

TABLE VI: Evolution of the peak ratio of the different algorithms for the function five uneven peak trap

Generation	0	10	25	60	175	350	425	500
PNA-NSGAII	0	0	0	0.11	0.66	**1**	**1**	0.98
MOBiDE	0	0	0	0.26	**1**	**1**	**1**	**1**
NSGAII	0	0.05	0.5	0.5	0.5	0.5	0.5	0.5
MOMMOPS	**0.97**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

TABLE VII: Evolution of the peak ratio of the different algorithms for the function modified rastringin

Generation	0	20	50	120	350	700	850	1000
PNA-NSGAII	**0**	0.01	0.05	0.09	0.09	0.08	0.24	0.98
MOBiDE	**0**	0.05	**0.53**	**0.61**	0.73	0.8	0.82	0.83
NSGAII	**0**	**0.12**	0.08	0.08	0.08	0.08	0.08	0.08
MOMMOPS	**0**	0	0	0	**1**	**1**	**1**	**1**

TABLE VIII: Evolution of the peak ratio of the different algorithms for the function CF3 in 3 dimension

Generation	0	26	66	159	466	933	1133	1333
PNA-NSGAII	**0**	0	0	0	0.05	0.15	**0.28**	**0.43**
MOBiDE	**0**	0	0	0.01	0.16	**0.17**	0.18	0.19
NSGAII	**0**	**0.09**	**0.17**	**0.17**	**0.17**	**0.17**	0.17	0.17
MOMMOPS	**0**	**0.09**	**0.17**	**0.17**	**0.17**	**0.17**	0.17	0.17

Access to our source code and to the results on all functions of the CEC2013 [14] benchmark is provided on our website[2].

A. Comparison of MOBiDE, PNA-NSGAII, MOMMOP and NSGAII

Let's first focus on the experiment comparing the different algorithms. In this experiment, the value of δ has been fixed to $0.9e^{-6}$ for MOBiDE. As we can see in Tab. X, the Kruskal-Wallis test often fails on the first few generations. But comparisons are relevant for all presented functions from generation 31.

We can observe in Tab. VI to IX that the original NSGAII has a fairly fast convergence, but its results are poor in term of peak ratio. Indeed, when looking at the detailed results, it appears that NSGAII is always retrieving exactly one optimum per run whereas other algorithms are retrieving more solutions for each function. This shows that the niching techniques and other mechanisms used in the dedicated multimodal algorithms

TABLE IX: Evolution of the peak ratio of the different algorithms for the function CF3 in 10 dimension

Generation	0	8	20	48	140	280	340	400
PNA-NSGAII	**0**	**0**	**0**	**0**	0	0	0.01	0.1
MOBiDE	**0**	**0**	**0**	**0**	0	0	0	0
NSGAII	**0**	**0**	**0**	**0**	**0.17**	**0.17**	**0.17**	0.17
MOMMOPS	**0**	**0**	**0**	**0**	0	0.05	0.06	**0.18**

TABLE X: Kruskal-Wallis results for algorithms comparison

Function name	Generations for which the Kruskal-Wallis test fails
Equal Maxima	0 to 1
Modified Rastringin All	0 to 2
CF3 3D	0 to 17
CF3 10D	0 to 31

[2]http://denispallez.i3s.unice.fr/doku.php?id=soft_multimodal

TABLE XI: Evolution of the peak ratio with different values of delta for the function five uneven peak trap

Generation	0	10	25	60	175	350	425	500
MOBiDE $0.9e^{-3}$	**0**	**0**	0.01	0.01	0.04	0.05	0.05	0.05
MOBiDE $0.9e^{-4}$	**0**	**0**	0.06	0.23	0.27	0.28	0.28	0.28
MOBiDE $0.9e^{-5}$	**0**	**0**	0.16	**1**	**1**	**1**	**1**	**1**
MOBiDE $0.9e^{-6}$	**0**	**0**	**0.26**	**1**	**1**	**1**	**1**	**1**

TABLE XII: Evolution of the peak ratio with different values of delta for the function modified rastringin

Generation	0	20	50	120	350	700	850	1000
MOBiDE $0.9e^{-3}$	**0**	0.03	0.33	0.63	0.84	0.88	0.89	0.9
MOBiDE $0.9e^{-4}$	**0**	**0.07**	**0.55**	**0.72**	**0.88**	**0.95**	**0.97**	**0.98**
MOBiDE $0.9e^{-5}$	**0**	0.04	0.54	0.62	0.78	0.85	0.87	0.87
MOBiDE $0.9e^{-6}$	**0**	0.05	0.53	0.61	0.73	0.80	0.82	0.83

improve performances when solving multimodal problems. However, if only one solution is required, the choice of the raw NSGAII seems to be the best. It converges faster or at the same speed than the others and always finds one optimum to the function.

Comparing the multimodal algorithms, we can observe that MOMMOP's convergence is not stable. It converges extremely fast for five uneven peak trap (Tab. VI) and Composition Function 3 (CF3) in 3 dimension (Tab. VIII), whereas its convergence is slow for modified rastringin (Tab. VII) and CF3 in 10 dimensions (Tab. IX). MOBiDE and PNA-NSGAII always present the same convergence scheme. MOBiDE tends to find a lot of solutions at the beginning of the run and then slowly improves. On the contrary, PNA-NSGAII has poor results at the beginning of the run but presents a big step in the retrieved results at around 75% of the run completion. This convergence scheme is probably due to the adaptive constraint that evolves exponentially as the run progresses.

Speaking of the peak ratio, MOMMOP is ahead for five uneven peak trap, modified rastringin and CF3 in 10 dimension. However, MOBiDE and PNA-NSGAII are not far behind. PNA-NSGAII gets the best results when dealing with CF3 in 3 dimension, retrieving more than twice as much optima as MOMMOP or MOBiDE. But this is the only case where this algorithm takes the lead. In other cases, it has lower or equal results than MOMMOP or MOBiDE and has a slower convergence. Therefore, MOMMOP and MOBiDE will most of the time be preferred.

When dealing with higher dimensional problems, like CF3 in 10 dimensions, MOMMOP stay stable in its results while MOBiDE completely fails. So, in the end, MOMMOP seems to be a good choice when dealing with MMOP today. It converges sometimes really fast and sometimes it is fairly slow, but never slower than PNA-NSGAII. Its performances are better or equal to those of MOBiDE and PNA-NSGAII and it is able to perform at least equally with NSGAII when dealing with composition functions. However, MOBiDE's results are not too bad and its stable and quick convergence can be preferred in some cases when we want to decrease the number of function evaluations. However MOBiDE depends on a parameter δ that needs to be setup manually and for which we don't have many information. That's why we explore the influence of this parameter on MOBiDE's performances in the next subsection.

B. *Influence of δ in MOBiDE*

In the whole previous experiment, the value of δ was constant. As we explained before, δ is a parameter of the niching technique introduced in MOBiDE. We decided to explore the influence of this parameter on the performances of MOBiDE. Indeed, MOBiDE presents good performances in most cases, but if it is too sensitive to its δ parameter, its use might be tricky in some real cases. Experiments have been done with δ ranging from $0.9e^{-6}$ to $0.9e^{-3}$. The Kruskal-Wallis tests in Tab. XIV show that no trusted comparison can be observed on the first generations, up to generation 201 for Vincent in 3 dimensions. However most of the presented data can be considered for comparisons safely.

In Tab. XI to XIII, we can see that depending on the benchmark function, the best results are not obtained with the same value of δ. For Vincent3D, the best value of δ is by far $0.9e^{-3}$, whereas for FiveUnevenPeakTrap the best value of δ would be by far $0.9e^{-5}$ or $0.9e^{-6}$. And for modified rastringin, the value of δ is of less importance. So we can conclude that δ is a sensitive and crucial parameter of MOBiDE which must be chosen carefully depending on the problem. We tried to infer some heuristic to set this parameter to its optimal value depending on the problem but were not able to come up with a satisfying solution yet. This makes MOBiDE a difficult method to tune, and its convergence speed strength is greatly balanced by the drawback of tuning δ for each specific problem.

TABLE XIII: Evolution of the peak ratio with different values of delta for the function Vincent 3D

Generation	0	26	66	159	466	933	1133	1333
MOBiDE $0.9e^{-3}$	**0**	0.01	0.19	0.30	0.42	0.55	0.60	0.63
MOBiDE $0.9e^{-4}$	**0**	0.02	0.20	0.29	0.33	0.38	0.39	0.40
MOBiDE $0.9e^{-5}$	**0**	0.02	0.19	0.29	0.32	0.34	0.35	0.36
MOBiDE $0.9e^{-6}$	**0**	**0.05**	**0.53**	**0.61**	**0.73**	**0.80**	**0.82**	**0.83**

TABLE XIV: MOBiDE influence of δ, Kruskal-Wallis results

Function name	Generations for which the Kruskal-Wallis test fails
Vincent 3D	0 to 201
Modified Rastringin all	0 to 24
Five Uneven Peak Trap	0 to 22

V. Conclusion

This work presented an extensive comparison of three recent multiobjective evolutionary algorithms: PNA-NSGAII, MOBiDE and MOMMOP. A fair comparison using the same benchmark functions and the same parameters has been conducted, including the study of the convergence scheme of each algorithm. We also analyzed the influence of the parameter δ of MOBiDE.

We observed that MOMMOP presents the best results at the end of the run, but PNA-NSGAII and MOBiDE are close to its performances. However, beside those performances, MOMMOP does not present a stable convergence scheme. It can be really fast or rather slow depending on the problem. Therefore, despite having a bit worse results, MOBiDE might be preferred in some cases because its convergence is always fast.

However, we also revealed that δ is a crucial parameter for MOBiDE and no heuristic has been found so far to set this parameter to an optimal value given a specific problem. This makes MOBiDE practically hard to use in real cases.

Finding an self-adaptive scheme for δ in MOBiDE would make MOBiDE a good concurrent method to MOMMOP. We would have two complementary methods in the field of multimodal evolutionary algorithms: one that converges faster but with less solutions and one that finds more solutions but might be slower. Both being interesting depending on the case.

It is worth mentioning that none of these algorithms were able to properly solve complex problems, introduced in the form of composition functions in the CEC2013 benchmark. So we should look closely to the apparitions of future competitors to be added to this benchmark in the future.

VI. Acknowledgment

This work is partly funded by French National Agency for Research, VISIIR project, under contract number ANR-13-CORD-0009.

References

[1] M. Preuss and S. Wessing, "Measuring multimodal optimization solution sets with a view to multiobjective techniques," in *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV*, ser. Advances in Intelligent Systems and Computing, M. Emmerich, A. Deutz, O. Schuetze, T. Bäck, E. Tantar, A.-A. Tantar, P. D. Moral, P. Legrand, P. Bouvry, and C. A. Coello, Eds. Springer International Publishing, 2013, vol. 227, pp. 123–137.

[2] K.-C. Wong, C.-H. Wu, R. K. Mok, C. Peng, and Z. Zhang, "Evolutionary multimodal optimization using the principle of locality," *Information Sciences*, vol. 194, no. 0, pp. 138 – 170, 2012, intelligent Knowledge-Based Models and Methodologies for Complex Information Systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025511006657

[3] O. Shir, "Niching in evolutionary algorithms," in *Handbook of Natural Computing*, G. Rozenberg, T. Bäck, and J. Kok, Eds. Springer Berlin Heidelberg, 2012, pp. 1035–1069.

[4] S. Das, S. Maity, B.-Y. Qu, and P. Suganthan, "Real-parameter evolutionary multimodal optimization — a survey of the state-of-the-art," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 71 – 88, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S221065021100023X

[5] S. Bandaru and K. Deb, "A parameterless-niching-assisted bi-objective approach to multimodal optimization," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*, June 2013, pp. 95–102.

[6] A. Basak, S. Das, and K. Tan, "Multimodal optimization using a biobjective differential evolution algorithm enhanced with mean distance-based selection," *Evolutionary Computation, IEEE Transactions on*, vol. 17, no. 5, pp. 666–685, Oct 2013.

[7] Y. Wang, H.-X. Li, G. Yen, and W. Song, "Mommop: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems," *Cybernetics, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[8] M. Jamil and X. Yang, "A literature survey of benchmark functions for global optimization problems," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, no. 2, pp. 150–194, 2013. [Online]. Available: http://arxiv.org/abs/1308.4008

[9] "A java-based evolutionary computation research system, ecj," http://cs.gmu.edu/∼eclab/projects/ecj/, accessed: 2015-02-04.

[10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *Trans. Evol. Comp*, vol. 6, no. 2, pp. 182–197, Apr. 2002. [Online]. Available: http://dx.doi.org/10.1109/4235.996017

[11] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[12] R. B. Agrawal, K. Deb, K. Deb, and R. B. Agrawal, "Simulated binary crossover for continuous search space," Tech. Rep., 1994.

[13] K. Deb and D. Kalyanmoy, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[14] X. Li, A. Engelbrecht, and M. G. Epitropakis, "Benchmark functions for cec'2013 special session and competition on niching methods for multimodal function optimization," Evolutionary Computation and Machine Learning Group, RMIT University, Australia, Tech. Rep., 2013. [Online]. Available: http://goanna.cs.rmit.edu.au/ xiaodong/cec13-niching/competition/cec2013-niching-benchmark-tech-report.pdf

[15] H. Coolican, *Research Methods and Statistics in Psychology*, ser. A Hodder Arnold Publication. Hodder & Stoughton, 2004. [Online]. Available: http://books.google.fr/books?id=RIaYQgAACAAJ