A Study of the Combination of Variation Operators in the NSGA-II Algorithm

Antonio J. Nebro[1], Juan J. Durillo[2], Mirialys Machín[3], Carlos A. Coello Coello[4], and Bernabé Dorronsoro[5]

[1] Department of Computer Science, University of Málaga, Spain
[2] Institute of Computer Science, University of Innsbruck, Austria
[3] Departamento de Computación, University of Informatic Sciences, Cuba
[4] CINVESAV-IPN, Mxico
[5] Computer Science Laboratory of Lille, Université Lille 1, France
antonio@lcc.uma.es, juan@dps.uibk.ac.at, mmachin@cav.uci.cu, ccoello@cs.cinvestav.mx, bernabe.dorronsoro_diaz@inria.fr

Abstract. Multi-objective evolutionary algorithms rely on the use of variation operators as their basic mechanism to carry out the evolutionary process. These operators are usually fixed and applied in the same way during algorithm execution, e.g., the mutation probability in genetic algorithms. This paper analyses whether a more dynamic approach combining different operators with variable application rate along the search process allows to improve the static classical behavior. This way, we explore the combined use of three different operators (simulated binary crossover, differential evolution's operator, and polynomial mutation) in the NSGA-II algorithm. We have considered two strategies for selecting the operators: random and adaptive. The resulting variants have been tested on a set of 19 complex problems, and our results indicate that both schemes significantly improve the performance of the original NSGA-II algorithm, achieving the random and adaptive variants the best overall results in the bi- and three-objective considered problems, respectively.

Keywords: Multiobjective Optimization, Evolutionary Algorithms, Variation Operators, Adaptation.

1 Introduction

Evolutionary algorithms (EAs) are a family of stochastic search techniques within metaheuristics [1] widely used on optimization. Genetic Algorithms (GAs), Evolution Strategies (ES), Genetic Programming (GP), and Differential Evolution (DE), among others, are examples of EAs. Specialized versions of EAs to solve multi-objective optimization problems usually referred as to MOEAs.

Most of EAs and MOEAs operate under a common principle: one or several individuals undergo the effect of some variation operators. Examples of these operators are the crossover and mutation operators, in the context of GAs, or the differential evolution operator in DE methods.

C. Bielza et al. (Eds.): CAEPIA 2013, LNAI 8109, pp. 269–278, 2013.

Some researchers have shown that some operators are more suitable for some types of problems than others. If we focus on multi-objective optimization, we can find some examples. Deb *et al.* evaluated the behavior of a number of operators for solving problems with variable linkages [3], and observed that the SBX operator was unable to deal with these types of problems. Iorio and Li [7] discussed the suitability of a number of operators for solving rotated problems and those having epistatic interactions among decision variables.

To make things harder, there is no reason to think that a variation operator is equally effective, in terms of its *evolvability* or ability to produce better solutions, over the whole search space of a given problem. In fact, the search space of real-world optimization problems may not be free of variable-linkage, epistasis, rotation, or complex relationships among their decision variables. Under these circumstances, the use of methods that keep their variation operators invariant through the whole execution of the EA may not be the best alternative.

Our goal is to investigate, in the context of multi-objective optimization, whether the combined use of different variation operators during the search may improve the performance of classical MOEAs. Our hypothesis is that the variation operators used in most of these algorithms can be effective in the exploration of certain regions of the search space of a given problem, but not over the whole search space. We study this idea by endowing NSGA-II with the ability to select its variation operators from a set containing different alternatives. The resulting algorithms are evaluated by solving problems with difficult Pareto sets; in particular, the LZ09 [9] benchmark and problems of the CEC 2009 competition [12]).

In this paper we propose two new versions of the NSGA-II algorithm which are able to select from among different variation operators during the search. We have considered a set composed of three operators commonly used in multi-objective optimization metaheuristics: SBX crossover, polynomial-based mutation, and the variation operator used in DE. The first proposed version of NSGA-II, referred to as NSGA-IIr hereinafter, creates new solutions by randomly selecting an operator from the set. The second version, named NSGA-IIa from now on, uses a record of the contribution of each operator in the past for selecting the operator to apply. This second scheme is based on the one proposed in the AMALGAM algorithm [11], and the idea is to give to these operators a higher probability of being chosen when they are capable of producing solutions that survive from one generation to the next. Additionally, we include in the study a version of NSGA-II using only the DE operator.

The rest of this paper is organized as follows. Next section reviews related work. Section 3 details our proposals. The methodology used in this work is described in Section 4 and the obtained results are analyzed in Section 5. Finally, we present our main conclusions and some possible paths for future research.

2 Previous Related Work

In this section we review existing works related to ours. We focus only on multi-objective optimization aproaches.

In [10], Toscano and Coello dealt with the issue of selecting the best operator for solving a given problem. These authors proposed a micro genetic algorithm, called μGA2, which runs several simultaneous instances of μGA2 configured with different variation operators. Periodically, the instance with the poorest performance was replaced with the best performing one. Thus, after several generations, all the parallel instances worked only with the best performing operator. A disadvantage of this approach is that once an operator had been discarded, it could not be used again in the execution of the algorithm.

MOSaDE [5] combines the use of four different versions of the DE operator. This combination was made in an adaptive way: the version that contributes the most to the search was given a higher probability of being used for creating new solutions. This contribution was measured by considering the success, in terms of the non-dominated solutions that it produced in the last n iterations of the algorithm. An improved version of MOSaDE with object-wise learning strategies, called OW-MOSaDE [6], participated in the CEC2009 MOEA competition [13], obtaining an average rank of 9.39 among 13 algorithms.

Vrugt and Robinson proposed in [11] the AMALGAM algorithm, based on the idea of using a number of multi-objective algorithms within a master algorithm. By measuring the contribution of each method in the last iteration, each algorithm was adaptively used favoring those techniques exhibiting the highest reproductive success. The algorithms used were NSGA-II, a PSO approach, a DE approach, and an adaptive metropolis search (AMS) approach.

Relate works propose therefore new algorithms or the combination of several existing techniques using a master approach, like in AMALGAM. Additionally, all of them use an scheme based on the contribution of the different operators for considering their application. The main point of our work, however, is not to propose a new algorithm but to analyze whether the combination of operators can improve the performance of an existing algorithm such as NSGA-II, when dealing with difficult multi-objective optimization problems.

3 NSGA-II with Combined Operators

This section aims at describing NSGA-IIa and NSGA-IIr. For the sake of clarity, we first present the original technique and then our proposals.

3.1 NSGA-II

NSGA-II (Deb *et al.* [2]) is the most popular multi-objective metaheuristic by far. It is a generational GA, so it is based on a population P of size n which, at each iteration, is used to create another population of n new solutions as follows. For every solution in P, two parents are selected and combined using the recombination operator and the result is later altered by means of a mutation operator. We use SBX crossover and polynomial mutation, as done in NSGA-II when adopting real-numbers encoding. As a result of these two operations, a new individual is created and inserted into a temporal population Q. Finally, P and

Algorithm 1. Pseudocode of NSGA-IIr.

```
1: Input: n // the population size
2: P ← Random_Population() // P = population
3: Q ← ∅                          // Q = auxiliar population
4: while not Termination_Condition() do
5:    for i ← 1 to (n) do
6:       randValue←rand();
7:       if (randValue ≤ 1/3) then
8:          parent←Selection1(P); // only one parent is selected
9:          offspring←PolynomialMutation(parent);
10:       else
11:          if (randValue ≤ 2/3) then
12:             parents←Selection2(P); // two parents are selected
13:             offspring←SBX(parents);
14:          else
15:             parents←Selection3(P); // three parents are selected
16:             offspring←DE(population[i], parents);
17:          end if
18:       end if
19:       Evaluate_Fitness(offspring);
20:       Insert(offspring,Q);
21:    end for
22:    R ← P ∪ Q
23:    Ranking_And_Crowding(R);
24:    P ← Select_Best_Individuals(R)
25: end while
26: Return P;
```

Q are merged in a single population R. The n best individuals, after applying the ranking and crowding procedures in R, will be selected to be the population P in the next generation of the algorithm. See further details in [2].

3.2 NSGA-IIr

NSGA-IIr is an extension of NSGA-II that makes use of three different variation operators: SBX crossover, polynomial mutation, and DE's variation operator. These operators are randomly selected whenever a new solution is to be produced. The pseudocode of this version is detailed in Algorithm 1.

The main difference with respect to the original NSGA-II lies in the parents selection mechanism and in the way in which offsprings are produced (lines 6-18). NSGA-IIr proceeds as follows. For each individual in P, it produces a random value in $[0, 1]$ (line 6). Depending of this value, one out of the three variation operators is selected, as shown in lines 7-18. Once the offspring is generated, the algorithm behaves as the original NSGA-II.

3.3 NSGA-IIa

NSGA-IIa applies the same variation operators as NSGA-IIr, but in an adaptive way, by taking into account their contribution, i.e., each operator selection probability is adjusted by considering that operator success in the last iteration.

The adaptive scheme considered for operator selection is based on the one used in AMALGAM [11]. Algorithm 2 describes such scheme. Assuming a number of $NumOperators$ different operators, the method computes the contribution of

Algorithm 2. Computing the contribution of each operator.

```
1: Input: P // population for the next iteration
2: total_contribution ← 0
3: for 1 ≤ operator ≤ NumOperators do
4:     contribution_operator ← solutionsInNextPopulation(operator,P) ;
5:     if contribution_operator ≤ threshold then
6:         contribution_operator ← threshold;
7:     end if
8:     total_contribution ← total_contribution + contribution_operator;
9: end for
10: for 1 ≤ operator ≤ NumOperators do
11:     probability_operator ← contribution_operator / total_contribution;
12: end for
```

all of them (loop between lines 3-9). The idea is to count how many solutions generated by each operator are part of the population P of the next generation (line 4). If an operator has contributed with less solutions than a minimum threshold, its contribution is set to this minimum threshold (lines 5-7); by doing so we avoid any operator to be discarded when producing no solutions in an iteration. Our motivation is that this operator may be useful later in a different phase of the search. In this work we have considered a threshold equal to 2, which was the value used in AMALGAM. Once the contribution of the operators have been computed, the probability of selecting them is updated (line 12). This way, the operators have a probability of being selected in the next generation which is proportional to their contribution.

3.4 NSGA-IIde

As we are using the DE operator in NSGA-IIr and NSGA-IIa, we consider interesting to include in the study another NSGA-II variant, where the mutation and crossover operators have been replaced by the DE operator. We have named this version NSGA-IIde. This way, we will have more information to determine if the performance improvements are not related to the use of a particular operator but to the combination of some of them.

4 Experimentation

Here we present the benchmark problems adopted for our tests, together with the parameter settings and the methodology followed in our experiments.

Benchmark Problems. We consider the LZ09 [9] benchmark and the problems defined for the CEC2009 competition [12]. The former is composed by nine problems (LZ09_F1 - LZ09_F9), all of which are bi-objective, except for LZ09_F6, which has three objectives. The latter contains problems with two, three, and five objectives, as well as constrained and unconstrained problems. We have selected the seven UF1- UF7 bi-objective and the UF8 - UF10 three-objective unconstrained problems.

Parameters Settings. For NSGA-II and its three variants we have used the same settings. The population size is 100, the SBX and polynomial mutation probabilities are 0.9 and $1/L$ (L is the number of decision variables of the problem being solved), respectively. Both operators share the same distribution index value, which is set to 20. The DE operator variant is current/1/bin, and the values of the CR and F control parameters are, respectively, 1.0 and 0.5. The stopping condition is $150,000$ function evaluations in the case of the LZ09 problems, and $300,000$ for the CEC 2009 problems.

Quality Assessment. To assess the performance of the algorithms we adopt two widely used indicators: additive epsilon [8]) and hypervolume [14].

Analysis of Results. For each combination of algorithm and problem we have made 30 independent runs, and we report the median, $\tilde{x}$, and the interquartile range, IQR, as measures of location (or central tendency) and statistical dispersion, respectively, for each considered indicator. When presenting the obtained values in tables, we emphasize with a dark gray background the best result for each problem, and a clear grey background is used to indicate the second best result; this way, we can see at a glance the most salient algorithms.

When comparing the values yielded by two algorithms on a given problem, we check if differences in the results are statistically significant. To cope with this issue, we have applied the unpaired Wilcoxon rank-sum test, a non-parametric statistical hypothesis test, which allows us to make pairwise comparisons between algorithms to analyze the significance of the obtained data [4]. A confidence level of 95% (i.e., significance level of 5% or p-value under 0.05) has been used in all cases, meaning that the differences were unlikely occurred by chance with a probability of 95%.

5 Comparison of Results

In this section, we analyze the obtained results when running the algorithms under the aforementioned experimental methodology. We first analyze the values yielded by the I_{ϵ}^{+} indicator, and then the ones obtained by the I_{HV} one.

The values obtained by the I_{ϵ}^{+} are summarized in Table 1. We start by analyzing the values obtained in the LZ09 family. As we can observe, the algorithm applying the adaptive combination of several operators, NSGAIIa, has led to an improvement of the results of the original version of NSGA-II in all the problems that are part of this benchmark, but it was outperformed, in turn, by the random variant, NSGA-IIr, in all the problems but two (LZ09_F4 and LZ09_F6). The NSGA-II variant using DE only achieved the best result in the first problem. Our wilcoxon analysis has relevealed that statistical significance has been found when comparing the the two extensions of NSGA-II algorithm with the original. Regarding to the comparison between our two proposals, there is no statistical significance in problems LZ09_F1 and LZ09_F4, NSGA-IIa outperforms NSGA-IIr in six problems, and NSGA-IIr improves NSGA-IIa in LZ09_F6, the only three-objective problem of the benchmark.

Table 1. LZ09 benchmark. Median and interquartile range of the I_ϵ^+ indicator.

	NSGA-II	NSGAII-r	NSGAII-a	NSGAII-de
LZ09_F1	$1.69e-02_{1.7e-03}$	$1.52e-02_{2.4e-03}$	$1.52e-02_{2.9e-03}$	$1.46e-02_{3.9e-03}$
LZ09_F2	$1.70e-01_{2.5e-02}$	$7.43e-02_{1.7e-02}$	$9.62e-02_{2.8e-02}$	$1.49e-01_{3.6e-02}$
LZ09_F3	$1.12e-01_{2.3e-02}$	$4.78e-02_{2.1e-02}$	$7.84e-02_{1.5e-02}$	$1.20e-01_{2.6e-02}$
LZ09_F4	$1.38e-01_{2.0e-02}$	$5.44e-02_{1.7e-02}$	$5.16e-02_{1.9e-02}$	$1.13e-01_{2.3e-02}$
LZ09_F5	$1.09e-01_{3.1e-02}$	$6.54e-02_{3.7e-02}$	$8.29e-02_{2.9e-02}$	$1.21e-01_{1.8e-02}$
LZ09_F6	$2.75e-01_{4.0e-02}$	$2.69e-01_{1.3e-02}$	$2.31e-01_{5.3e-02}$	$6.38e-01_{2.4e-01}$
LZ09_F7	$3.32e-01_{1.7e-01}$	$3.48e-02_{2.1e-02}$	$1.28e-01_{1.4e-01}$	$1.00e+00_{0.0e+00}$
LZ09_F8	$2.76e-01_{1.4e-01}$	$2.22e-01_{6.7e-02}$	$2.54e-01_{1.5e-01}$	$9.10e-01_{2.1e-01}$
LZ09_F9	$1.87e-01_{6.5e-02}$	$7.98e-02_{3.7e-02}$	$1.05e-01_{2.3e-02}$	$1.49e-01_{3.2e-02}$
UF1	$1.54e-01_{2.4e-02}$	$1.82e-02_{2.7e-03}$	$5.51e-02_{2.6e-02}$	$1.30e-01_{2.8e-02}$
UF2	$9.35e-02_{2.5e-02}$	$5.53e-02_{2.4e-02}$	$6.73e-02_{3.1e-02}$	$1.13e-01_{2.8e-02}$
UF3	$3.12e-01_{1.1e-01}$	$4.63e-02_{6.0e-02}$	$1.40e-01_{1.1e-01}$	$2.35e-01_{4.7e-02}$
UF4	$4.95e-02_{2.5e-03}$	$4.62e-02_{1.8e-03}$	$5.11e-02_{5.1e-03}$	$8.27e-02_{9.8e-03}$
UF5	$3.80e-01_{7.2e-02}$	$4.25e-01_{2.0e-01}$	$5.00e-01_{2.1e-01}$	$1.05e+00_{3.5e-01}$
UF6	$3.68e-01_{1.5e-01}$	$4.62e-01_{3.9e-01}$	$5.30e-01_{3.0e-01}$	$4.11e-01_{1.9e-01}$
UF7	$1.33e-01_{3.5e-01}$	$5.68e-02_{1.8e-02}$	$7.55e-02_{3.4e-02}$	$9.68e-02_{2.7e-02}$
UF8	$3.05e-01_{4.4e-01}$	$7.13e-01_{4.5e-01}$	$3.09e-01_{4.5e-01}$	$8.56e-01_{1.7e-01}$
UF9	$4.91e-01_{2.8e-01}$	$5.57e-01_{4.0e-01}$	$4.58e-01_{2.1e-01}$	$8.65e-01_{2.7e-01}$
UF10	$9.31e-01_{1.3e-01}$	$9.44e-01_{1.6e-01}$	$8.61e-01_{1.5e-01}$	$1.97e+00_{6.4e-01}$

Table 2. LZ09 benchmark. Median and interquartile range of the I_{HV} indicator.

	NSGA-II	NSGA-IIr	NSGA-IIa	NSGA-IIde
LZ09_F1	$6.53e-01_{1.1e-03}$	$6.55e-01_{6.9e-04}$	$6.55e-01_{7.2e-04}$	$6.56e-01_{7.6e-04}$
LZ09_F2	$5.53e-01_{1.3e-02}$	$6.35e-01_{9.6e-03}$	$6.25e-01_{2.1e-02}$	$5.66e-01_{3.6e-02}$
LZ09_F3	$6.24e-01_{8.1e-03}$	$6.46e-01_{3.5e-03}$	$6.41e-01_{6.5e-03}$	$5.85e-01_{1.6e-02}$
LZ09_F4	$6.34e-01_{4.7e-03}$	$6.44e-01_{2.8e-03}$	$6.48e-01_{3.8e-03}$	$5.89e-01_{1.4e-02}$
LZ09_F5	$6.28e-01_{1.1e-02}$	$6.43e-01_{1.0e-02}$	$6.41e-01_{8.5e-03}$	$5.96e-01_{1.2e-02}$
LZ09_F6	$2.08e-01_{3.4e-02}$	$2.51e-01_{2.7e-02}$	$2.89e-01_{2.9e-02}$	$4.35e-02_{4.7e-02}$
LZ09_F7	$4.80e-01_{4.3e-02}$	$6.50e-01_{4.5e-03}$	$6.37e-01_{2.9e-02}$	$0.00e+00_{0.0e+00}$
LZ09_F8	$4.62e-01_{4.8e-02}$	$5.33e-01_{3.8e-02}$	$5.00e-01_{4.4e-02}$	$0.00e+00_{0.0e+00}$
LZ09_F9	$2.25e-01_{4.1e-02}$	$2.99e-01_{1.7e-02}$	$2.88e-01_{1.8e-02}$	$2.31e-01_{2.9e-02}$
UF1	$5.73e-01_{1.9e-02}$	$6.53e-01_{8.9e-04}$	$6.47e-01_{7.7e-03}$	$5.78e-01_{2.9e-02}$
UF2	$6.34e-01_{8.9e-03}$	$6.47e-01_{4.4e-03}$	$6.46e-01_{6.0e-03}$	$5.99e-01_{1.3e-02}$
UF3	$4.74e-01_{4.8e-02}$	$6.38e-01_{2.1e-02}$	$6.00e-01_{5.7e-02}$	$3.43e-01_{5.4e-02}$
UF4	$2.64e-01_{1.4e-03}$	$2.68e-01_{6.5e-04}$	$2.66e-01_{2.1e-03}$	$2.34e-01_{1.4e-02}$
UF5	$1.87e-01_{8.1e-02}$	$2.39e-01_{2.1e-01}$	$1.98e-01_{1.5e-01}$	$0.00e+00_{0.0e+00}$
UF6	$2.43e-01_{6.7e-02}$	$2.34e-01_{1.6e-01}$	$2.39e-01_{1.6e-01}$	$5.42e-02_{6.2e-02}$
UF7	$4.41e-01_{8.8e-02}$	$4.81e-01_{7.1e-03}$	$4.77e-01_{4.7e-03}$	$4.51e-01_{1.9e-02}$
UF8	$1.96e-01_{9.6e-02}$	$1.08e-01_{1.4e-01}$	$1.87e-01_{1.6e-01}$	$0.00e+00_{3.9e-04}$
UF9	$3.21e-01_{1.6e-01}$	$1.78e-01_{3.7e-01}$	$3.94e-01_{2.0e-01}$	$3.53e-02_{6.9e-02}$
UF10	$1.73e-02_{2.3e-02}$	$0.00e+00_{2.7e-02}$	$3.63e-02_{6.3e-02}$	$0.00e+00_{0.0e+00}$

Regarding the problems of the CEC 2009 competition we can see that the random NSGA-II variant achieved the best values in five out the seven bi-objective problems and no best results in the tree-objective instances. The applied Wilcoxon Rank-sum test showed, however, that the differences with NSGA-II in problems UF5, UF6, and UF8 were not statistically significant. NSGA-IIa performed better that NSGA-II in problems UF4, UF5, and UF6, and it outperformed NSGA-IIr in UF9 and UF10 with confidence in all these instances.

The values for the I_{HV} are included in Table 2. A simple comparison with the convergence indicator results (Table 1) shows almost an identical performance of the algorithms for the LZ09 benchmark; however, the Wilcoxon ranks-sum test values showed some differences. According to the I_{HV}, NSGA-IIa obtained a better value in LZ09_F4 with statistical confidence and the differences in LZ09_F5 are non significant. Some values in Table 2 are 0; this means that the approximation front produced by the algorithm was beyond the limits of the Pareto front used to calculate the I_{HV} indicator, so none of the solutions contribute to the

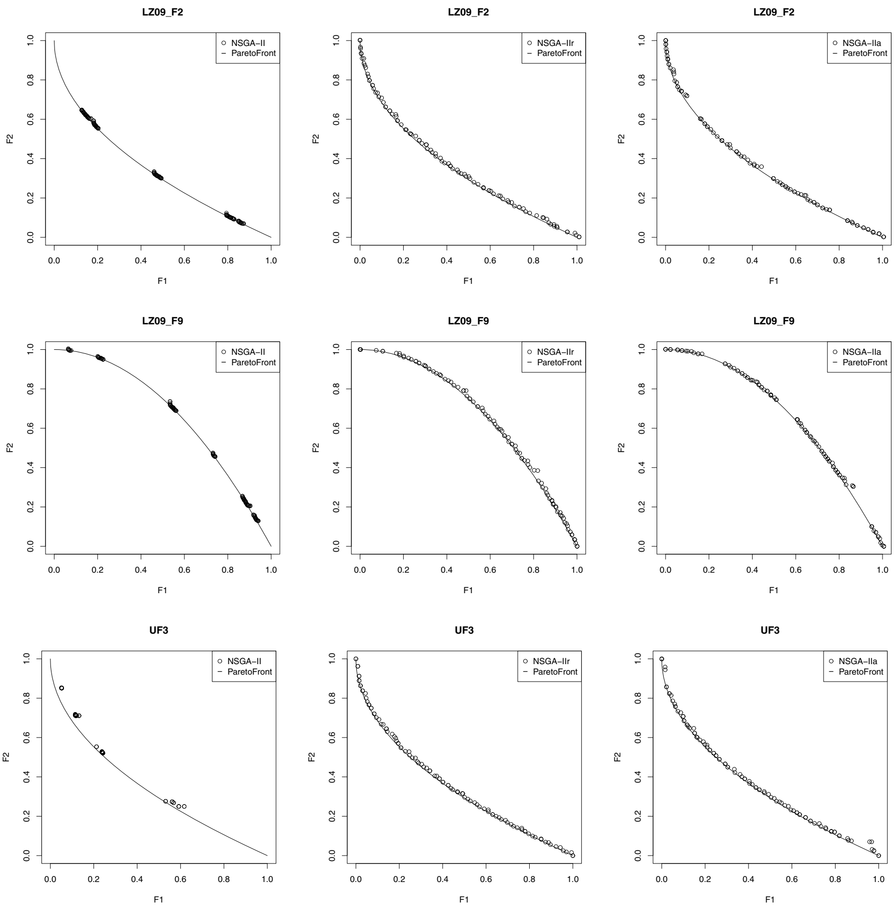

Fig. 1. Computed approximations for problems LZ09_F2, LZ09_F9, and UF3 with NSGA-II (left), NSGA-IIr (center), and NSGA-IIa (right)

hypervolume. In the case of the second evaluated benchmark, NSGA-IIr got again the best figures in most of the bi-objective problems, but it was outperformed by NSGA-II in the three-objective UF8, UF9, and UF10 instances by NSGA-II, being the differences significant according to the applied statistical methodoly. NSGA-IIa also yielded best results than NSGA-IIr in the same problems, although the differences are not significant in the UF8 problem. Compared with NSGA-II, NSGA-IIa obtained better values in six out the ten studied problems with confidence, being the differences in the rest of problems non significant.

To illustrate the performance of our proposals, we include the best Pareto front approximations found by the NSGA-II and its two variants according to the I_{HV} in Fig. 1 for problems LZ09_F2, LZ09_F9, and UF3. We can observe these problems posed a lot of difficulties to NSGA-II, which produced very poor approximation sets. The extensions of NSGA-II have generated better results in terms of the quality of the computed fronts, which can be visually stated.

6 Discussion

From the previous study we can infer some facts. First, it is clear that the combined used of the three chosen operators, in an adaptive or in a random way, lead to algorithms outperforming NSGA-II in most of the considered problems. Given that NSGA-IIde does not achieve better results compared with the original algorithm (with the exception of the LZ09_F1 problem) we conclude that the combination of the three operators is the reason of the performance improvements that are obtained by both the NSGA-IIr and NSGA-IIa variants.

The 19 evaluated problems have complex Pareto sets and most of state-of-the-art Pareto dominance-based MOEAs experiment troubles when solving them, so the enhancements illustrated by Fig. 1 are remarkable. Consequently, we infer that the combination of operators has a positive influence in the performance of the resulting algorithms, allowing a better exploration of the search space, thus supporting our initial hyphotesis of that the variation operators used in many MOEAs can be effective in the exploration of certain regions of the search space of a given problem but not over the whole search space.

Our analysis revealed that the random selection of operators provides overall better results than the adaptive version in bi-objective problems, while the latter outperforms the former in three-objective problems. This issue deserves further research.

7 Conclusions

We have studied two schemes for using variation operators in a combined way in the NSGA-II algorithm. The first one selects the operators at random, while the second one takes them in an adaptive way. The considered operators have been SBX crossover, polynomial mutation, and the DE operator. To assess the performance of the two combined strategies we have taken 19 multiobjective problems, two quality indicators, and we have statistically ensured the confidence of the obtained results. A version of NSGA-II using only the differential evolution operator has been included for completeness.

The experiments carried out revealed that the combinator of operators enhances the performance over the original NSGA-II algorithm. The random scheme was the most salient variant when solving the bi-objective problems, while the adaptive algorithm yielded the best results in the three-objective instances. The improvements achieved in many problems are remarkable; therefore, we conclude that the combined use of variation operators can improve classical MOEAs, as shown in the context of the experimentation carried out. It is worth noting that the modifications of the NSGA-II algorithm are kept in a minimum.

As future work, we plan the inclusion of a broader set of operators. The application of the analyzed variation schemes to other multi-objective evolutionary algorithms (e.g., MOEA/D), the study of potential benefits when applying it for solving scalable problems in the number of variables or objectives, and the investigation of why the random and adaptive schemes yield, respectively, the best Pareto front approximations in the bi- and three-objective selected problems are also a matter of future work.

Acknowledgments. Antonio J. Nebro is supported by projects TIN2011-25840 (Spanish Ministry of Education and Science) and P11-TIC-7529 (Innovation, Science and Enterprise Ministry of the regional government of the Junta de Andalucía). Juan J. Durillo acknowledges the Austrian Research Promotion Agency under contract nr. 834307 (AutoCore). Carlos A. Coello Coello acknowledges support from CONACyT project no. 103570. Bernabé Dorronsoro acknowledges that the present work is partially supported by the National Research Fund, Luxembourg, and cofunded under the Marie Curie Actions of the European Commission (FP7-COFUND).

References

1. Blum, C., Roli, A.: Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys 35(3), 268–308 (2003)
2. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE TEVC 6(2), 182–197 (2002)
3. Deb, K., Sinha, A., Kukkonen, S.: Multi-objective test problems, linkages, and evolutionary methodologies. In: GECCO 2006, pp. 1141–1148 (2006)
4. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7, 1–30 (2006)
5. Huang, V.L., Qin, A.K., Suganthan, P.N., Tasgetiren, M.F.: Multi-objective optimization based on self-adaptive differential evolution algorithm. In: Proceedings of the 2007 IEEE CEC, pp. 3601–3608 (2007)
6. Huang, V.L., Zhao, S.Z., Mallipeddi, R., Suganthan, P.N.: Multi-objective optimization using self-adaptive differential evolution algorithm. In: Proceedings of the 2009 IEEE CEC, pp. 190–194 (2009)
7. Iorio, A.W., Li, X.: Solving rotated multi-objective optimization problems using differential evolution. In: Australian Conference on Artificial Intelligence, pp. 861–872 (2004)
8. Knowles, J., Thiele, L., Zitzler, E.: A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. Technical Report 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich (2006)
9. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. IEEE TEVC 2(12), 284–302 (2009)
10. Toscano Pulido, G., Coello Coello, C.A.: The micro genetic algorithm 2: Towards online adaptation in evolutionary multiobjective optimization. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 252–266. Springer, Heidelberg (2003)
11. Vrugt, J.A., Robinson, B.A.: Improved evolutionary optimization from genetically adaptive multimethod search. Proceedings of the National Academy of Sciences of the United States of America 104(3), 708–711 (2007)
12. Zhang, Q., Suganthan, P.N.: Special session on performance assessment of multi-objective optimization algorithms/cec 09 moea competition (May 2009)
13. Zhang, Q., Zou, A., Zhao, S., Suganthan, P.N., Liu, W., Tivari, S.: Multiobjective optimization test instances for the cec 2009 special session and competition. Technical Report CES-491, School of CS & EE, University of Essex (April 2009)
14. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE TEVC 3(4), 257–271 (1999)