Alejandro Cervantes, David Quintana, Gustavo Recio. *Efficient dynamic resampling for dominance-based multiobjective*. To appear in "Engineering Optimization"
DOI: 10.1080/0305215X.2016.1187729

To appear in *Engineering Optimization*
Vol. 00, No. 00, Month 20XX, 1–19

Efficient Dynamic Resampling for Dominance-based Multiobjective Evolutionary Optimization

Alejandro Cervantes*, David Quintana, Gustavo Recio

Department of Computer Science, Universidad Carlos III de Madrid, Spain

(*April 27, 2016*)

Multi-objective optimization problems are often subject to the presence of objectives that require expensive resampling for their computation. This is the case of many robustness metrics, frequently used as an additional objective, that accounts for the reliability of specific sections of the solution space. Typical robustness measurements use resampling, but the number of samples that constitute a precise dispersion measure has a potentially large impact on the computational cost of an algorithm. This paper proposes the integration of dominance based statistical testing methods as part of the selection mechanism of MOEAs with the aim of reducing the amount of fitness evaluations. The performance of the approach is tested on five classical benchmark functions integrating it into two well-known algorithms NSGA-II and SPEA2. The experimental results show a significant reduction in the number of fitness evaluations while, at the same time, maintaining the quality of the solutions.

Keywords: Evolutionary Multi-objective Optimization; Uncertainty; Resampling

1. Introduction

A substantial portion of real-world optimization problems is multi-objective. That is, they require the simultaneous optimization of a number of objectives that are often in conflict. Therefore, it is no surprise that there is great interest in the development of algorithms that can deal with this feature. In the past decade and a half, there has been a growing interest on a family of algorithms called Evolutionary Multi-Objective Optimization Algorithms (MOOAs) (Coello 2000). Among them, there is an especially large subset of approaches that fall under the category of Evolutionary Multi-Objective Genetic Algorithms (MOEAs) (Coello Coello 2006). These have been successfully used to solve problems on a wide range of domains such as data mining, aerospace engineering or finance, to mention a few (Mukhopadhyay et al. 2014b,a; Arias-Montano, Coello Coello, and Mezura Montes 2012; Ponsich, Jaimes, and Coello 2013).

The difficulty in solving the above mentioned problem is sometimes increased by the presence of uncertainty. Uncertainties can affect different aspects of the problem such as the decision variables or the objective functions. Following the first efforts by Branke (1998) and his team, the challenge of adapting MOEAs to handle uncertain environments has recently grabbed the attention of researchers (Deb and Gupta 2006). Though there are some other potential strategies to deal with this problem (i.e. surrogate models) resampling seems to have gained the researchers attention. The basic idea behind resampling is that the analysis of the dispersion pattern on the objective space, originated

*Corresponding author. Email: acervant@inf.uc3m.es

as a consequence of gathering a pseudo-random set of samples within a neighborhood region (on the variable space) surrounding a given solution, or a component of the fitness function, will be a good indicator of its robustness. The sensitivity of the solution to the presence of perturbation can subsequently be used by the evolution process to drive the population towards stable or robust regions of the solution space.

Robustness is often considered through the evolution process as a decision component that favours solutions that are less sensitive to perturbations (Hassan and Clack 2008; García et al. 2012). Alternatively, it may be taken into account as an additional objective function, extending the problem with a new objective (Chicano et al. 2012). The former approach would be transparent to the decision maker, whereas the second, would allow the decision maker to evaluate the sensitivity of the solutions to perturbation of decision variables. Under this framework, the less sensitive ones would be considered better solutions. Even though the above approaches offer good results, they are computationally inefficient. The evaluation of objective functions is often expensive and considering robustness, as described above, is likely to result in more fitness computations than necessary.

This paper is aimed at introducing a way to reduce the number of evaluations needed in those circumstances, i.e. obtaining robust solutions. Many popular Multi-Objective Evolutionary Algorithms, such as NSGA-II (Deb et al. 2002), MOPSO (Moore and Chapman 1999), PAES (Knowles and Corne 1999) or SPEA2 (Zitzler, Laumanns, and Thiele 2001) evaluate candidate solutions in terms of dominance. Meaning that, whenever two solutions are compared, the exact value of each objective function is not as relevant as which solution, if any, dominates over the other. When considering robustness, the above dominance relation involves averaging the fitness of a fixed number of neighboring solutions. This work proposes the use of significance statistical tests to establish the above mentioned dominance relation using fewer evaluations of neighboring solutions. This approach is conceptually related to algorithms like RACE and BRACE (Moore and Lee 1994), Rational Allocation of Trials (Teller and Andre 1997) and some others in the context of dynamic resampling (Syberfeldt et al. 2010) or model selection (Giacobini, Tomassini, and Vanneschi 2002).

The rest of the paper is organized as follows. Section 2 provides the fundamental background on robust optimization. The optimization problem under uncertainty is characterized here and some general efforts to overcome the presence of noise in the decision variables are described. Then, a method to do efficient sampling is introduced in section 3. The proposed method will be tested on several well-known benchmark functions in section 4. Finally, section 5 deals with the summary, conclusions and future work.

2. Robust solutions in Multi-objective Problems

In many real world optimization problems multiple and often conflicting objectives are found. Thus, it is normal to look at them as a multi-objective optimization problem. A general multi-objective problem consists of finding a vector of decision variables which satisfies a set of constraints and optimizes a vector function whose elements represent the objective functions. These objective functions are in general in conflict with each other, therefore the solution to be found should give the values of all the objective functions acceptable to the designer (Osyczka 1984).

The vector function to be optimized can be defined as

$$\bar{f}(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), ..., f_k(\bar{x}))]^T \tag{1}$$

where $\bar{x} = [x_1, x_2, ..., x_n]^T$ is the vector of decision variables which optimizes the above

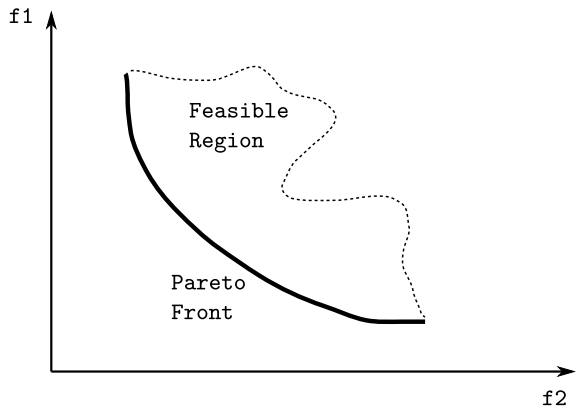


Figure 1. Example of an optimization problem with two objective functions. The Pareto front is depicted with a bold line at the inferior boundary of the feasible region.

function and satisfies the m inequality constraints

$$g_i(\bar{x}) \geq 0 \qquad i = 1, 2, ..., m \tag{2}$$

Thus, the general multiobjective problem could be described as finding, among all possible solutions that satisfy the above inequality, the particular set $x_1^*, x_2^*, ..., x_k^*$ which yields the optimum values of all the objective functions.

Therefore the optimizing process of multiobjective problems does not involve finding a single minima but a set of solutions that satisfies above formulation. Using equations (1) and (2), a point $\bar{x}^* \in \mathcal{F}$ is said to be Pareto optimal if for every $\bar{x} \in \mathcal{F}$ either,

$$(f_i(\bar{x}) = f_i(\bar{x}^*)) \qquad \text{for all} \quad i \in I \tag{3}$$

while there is at least one $i \in I$ such that

$$f_i(\bar{x}) > f_i(\bar{x}^*) \tag{4}$$

Above concept of Pareto optimum was formulated by Vilfredo Pareto in the XIX century (Pareto 1896), and constitutes by itself the origin of research in multiobjective optimization. Bearing on mind this concept, Pareto optimum does not mean a single solution, but rather a set of solutions called non-dominated solutions. The minima in this sense, are going to be in the boundary of the design region forming the Pareto front. Figure 1 represents the solution space for an optimization problem with two objectives, the Pareto front is depicted as a bold line. In general, it is not easy to find an analytical expression of the line or surface that contains these points, and the normal procedure is to compute the points $\mathcal{F}^k$ and their corresponding $f(\mathcal{F}^k)$ (Coello 1998). Typically, within the Pareto front a knee point is always a preferred trade-off solution (Deb and Gupta 2010).

The fact that evolutionary algorithms deal simultaneously with a set of possible solutions makes them particularly appropriate to solve multiobjective optimization problems. This allows finding an entire set of Pareto optimal solutions in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of the traditional mathematical programming techniques. Also mutiobjective optimization offers a unique way of analyzing the solutions to a problem. Since all solutions are optima, it is possible to analyze the trade off information of sacrificing one objective towards a gain in another. This allows getting information regarding the cost of a transition from one optimum non dominated solution to another. In addition to all of above, Deb (1999)

summarized the main disadvantages of classical optimization methods when solving multiobjective problems as:

(1) An algorithm has to be applied many times to find multiple Pareto-optimal solutions.
(2) Most algorithms demand some knowledge about the problem being solved.
(3) The spread of Pareto-optimal solutions depends on efficiency of the single objective optimizer.
(4) In problems involving uncertainties or stochasticities, classical methods are not reliable.
(5) Since classical single-objective optimizers are not efficient in handling discrete search space problems (Deb 1995; Deb and Goyal 1998), they will neither be efficient for multi-criterion optimization problems having discrete search space.

Concluding with the statement that all the above difficulties can be handled by using an evolutionary search algorithm. Additional information on evolutionary multiobjective algorithms can be found in Veldhuizen and Lamont (1998), Tamaki, Kita, and Kobayashi (1996), and Fonseca and Fleming (1995).

Additionally, many real world optimization problems are subject to uncertainties and noise. In practice, a solution cannot be physically implemented to the desired accuracy and the physical implementation of the solution may be somewhat different from the theoretical global optimal solution. If a global solution is quite sensitive to variable perturbation (or noise) in its vicinity, the implemented solution may result in a different set of objective values from those of the theoretical optimal solution.

In general, for real world problems there are four categories of uncertainties (Jin and Branke 2005): those where the fitness evaluation is subject to noise; changes that take place after the optimal solution has been determined which affect the design variables are considered as the second source of uncertainties (a solution should still work satisfactorily when the design variables change slightly); the use of models instead of precise data and the existence of measurement errors in the calculation of the system output leads to uncertainties in the objective function values; and finally, the fourth category of uncertainties is related to time-varying fitness functions (the fitness function is deterministic at any point in time, but is dependent on time). Even though the basic idea behind the approach discussed in this paper could be applicable in more scenarios, we will only consider uncertainties originated by variations in the variable space.

For this reason, emphasis must be made in finding robust solutions which are less sensitive to perturbations in their neighborhoods. If a robustness measure is taken on the solution, the general problem could be modelled as a multiobjective optimization problem which includes robustness as another objective function which has the desired effect over non dominated solutions. It may happen that a non dominated solution in the Pareto front is dominated by another solution in the robust Pareto front. In this case such solution is not of much interest anymore and it is possibly replaced by a non dominated solution. Figure 2 illustrates the above situation, there is a considerable chance that either B or C or both get dominated in the robust pareto front, whereas solution A is obviously more robust than the rest.

Further description of robust optimization methods can be found in Beyer and Sendhoff (2007), Marijt (2009) and Talbi (2009). In Deb and Gupta (2006) the authors propose two main methods to achieve robustness in the solutions. The first method replaces the objective function with a mean effective function. Whereas the second method calculates a normalised difference between the objective function value f and the perturbed function value f^p. In this paper a variation of the second method which includes confidence based dynamic resampling is proposed.

The procedure of computing the robustness value for a single sample solution by forcing

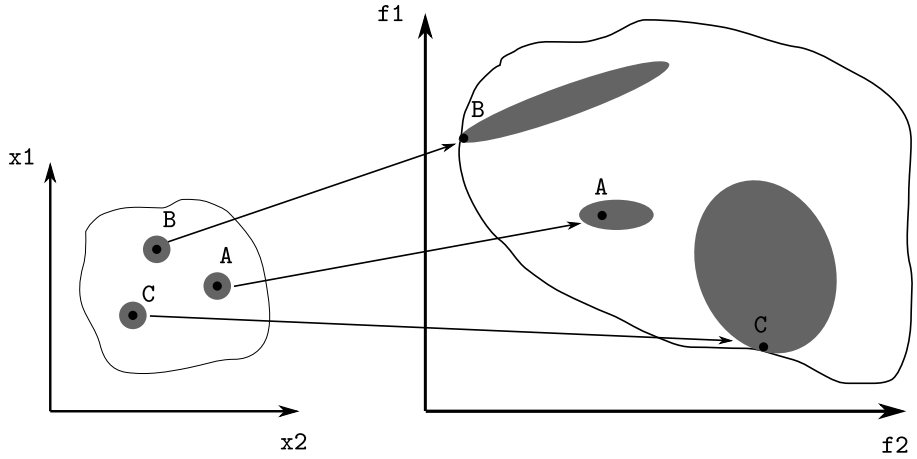


Figure 2. Robust Pareto front formation and corresponding change in the Pareto dominance. Solution A is far more robust as the effect of variations in the variable space is smaller when observed in the objective space.

a variation in the variable space is as follows. First a set of instances is generated around the original instance solution within the variable space (left hand side plot in Figure 2). For each instance of the generated cloud set, all the objective function values must be computed, this in turn generates a cloud set within the objective space (right hand side plot in Figure 2). The larger the generated cloud set the larger the computational requirement. Thus, including the computation of robustness into any multiobjective algorithm can be computationally very costly

Kruisselbrink, Emmerich, and Bäck (2010) suggest that this computational cost can be substantially reduced and introduce ABRSS, an scheme that enforces locally well-spread distributions of archive points. Saha, Ray, and Smith (2011) suggest controlling the number of solutions whose robustness is assessed, and they present IDEAR. This algorithm limits the size the growth of the archive; focuses the evaluation on relatively healthy individuals; and only evaluates fully the variations on solutions that are under-represented in the archive. Another potential way to overcome this problem consists on applying dynamic resampling and will be described in the next section.

The advantages of using this approach, that adapts the number of samples around each individual to the circumstances, have been previously discussed by Pietro, While, and Barone (2004), where the authors discuss two dynamic strategies to reduce noise. Siegmund, Ng, and Deb (2013) elaborate on the matter and present the main approaches together with the most important contributions in the area. This work was extended in Siegmund, Ng, and Deb (2015), where the authors suggest combining multiple resampling strategies to come up with systems that base the sampling allocation on multiple factors.

As is was mentioned in the introduction, this work is also related to a resampling strategy introduced by Syberfeldt et al. (2010) for MOPSA-ES. These authors rely on the same statistical test to drive the selection of the number of samples. However, the choices made to implement the concept result in differences like the number of robustness objectives handled; the easiness to adapt the concept to different MOOAs or their suitability with large populations, among others.

Further discussion on the dynamic resampling technique used through this paper can be found in the following section.

3. Efficient Resampling using Statistical Tests

The objective of this paper consists in introducing an efficient way to manage perturbation-based indicators in dominance-based multi-objective algorithms. As it was mentioned in the precedent section, these indicators are mostly used to model robustness. They can be either used to bias the behavior of the genetic algorithm, or to extend the original formulation adding additional objectives. The approach presented in this paper falls in the latter category.

The strategy of extending the model with a new objective comes at the price of requiring larger populations, which then results in higher computational costs. However, it has also the advantage of offering the decision maker valuable information on the reliability of the partial solutions that define the Pareto front.

Whenever two individuals of the population, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, need to be compared in terms of the dispersion-based objective during the optimization process, a neighborhood is defined for each of them. This area will be bounded to a parameter, $\boldsymbol{p}$, whose value is fixed and common to every solution. Then, the objective can be computed perturbing the solutions n times within the described limits.

In practice, the performance of the algorithm is often highly dependent on the number of samples, n. The more samples are used in this process, the better the accuracy of the dispersion estimate and, therefore, the more reliable are the comparisons based on it. However, the accuracy resulting from large samples comes at the price of larger computational cost due to the fact that every additional sample requires a new fitness evaluation. Therefore, there is a rational effort for trying to balance these two and limit as much as possible the number of evaluations, especially in those domains where fitness evaluations are very expensive. The strategy presented in this paper attempts to reduce the amount of sampling required for each candidate solution, keeping the fixed parameter n as an upper bound limit.

Given that in dominance-based algorithms, only the relative value of the objective is required during the optimization process, it is very likely that a fraction of the allowed samples might be enough to tell whether one individual dominates the other in terms of the dispersion objective. For this reason, we suggest the process that follows.

Initially, only two samples per individual are taken into account for the calculations, meaning that the initial dispersion is based on just two nearby samples and computed as the average distance from $\boldsymbol{f}(\boldsymbol{x})$ to $\boldsymbol{f}(\boldsymbol{x}'_i)$, being the subscript i the sample number. These estimates are then compared to identify which of the solutions is more robust.

Due to the stochastic nature of the sampling mechanism, the statistical significance of the average observed differences should be confirmed. There are different alternatives suitable to test whether average dispersion difference for samples related to each solution is statistically significant or not. Among them, the Welch test (Welch, B. L. 1947) is one of the most widely used. This test is an extension of Student's t-test of equal averages that does not require the samples involved to have the same variance. Formally, the test is based on the statistic.

$$t \quad = \quad \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{5}$$

where $\overline{X}_1$, s^2 and n represent the observed means, variances and sizes of the samples considered.

For testing purposes, the distribution of the defined statistic is approximated by Student's t distribution with *df* degrees of freedom.

$$df = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1 - 1} + \frac{\left(\frac{s_2^2}{n_2}\right)^2}{n_2 - 1}} \tag{6}$$

As it was mentioned, the test is very similar to Student's t-test but, apart from the differences in the computation of the degrees of freedom, it does not use a pooled variance.

If the null hypothesis of equal mean distance is rejected, we will admit that one individual dominates the other in terms of the dispersion objective used to model robustness (not necessarily as a solution, as the rest of the objectives should also be considered). Otherwise, more evidence is required.

In this case, another sample will be added to the robustness estimate of the solution with fewer samples (the choice will be at random when both estimates are worked out using the same number of samples), and the difference will be tested once more. The process of adding samples and testing for equal robustness will then be repeated until either the test rejects the hypothesis or the number of samples used to compute the estimates reaches the upper limit n. In the latter case, dominance is assigned by direct comparison of the robustness values.

Even though the number of evaluations potentially saved by the strategy depends on the structure of the fitness landscape, the upper bound in the worst-case scenario would require the same number of samples used by the standard static resampling method.

The choice of Welch test has a drawback that has not yet been mentioned, namely, a normality assumption that will not always hold. Under some circumstances, such as sufficient large samples or a directional use (as it is the case of the implementation in this work), results can reasonably cope with violations (Sawilowsky and Blair 1992) but, in general, the null hypothesis of equal means would be less likely to be rejected. In practice, this means that, under non-normality, more samples would be required to conclude that the robustness of a solution is significantly larger than that of another. This fact could reasonably be expected to limit the benefits of this approach. Non-parametric tests, such as the Wilcoxon Rank Sum test (Mann, H. B. and Whitney, D. R. 1947), are not constrained by this particular problem and, therefore, might seem to be better alternatives. However, the approach undertaken in this work is justified on grounds of scalability.

The idea of carrying over information related to the neighborhood of individuals has the potential to reduce the number of fitness evaluations. This is due to the fact that future comparisons involving these elements would not require a complete resampling, as the process could resume adding new evidence to the information already accumulated. However, it also brings up a decision problem: the nature and amount of data to be kept.

The extreme solution of saving all the fitness computations from the samples used to calculate the robustness indicator offers, on the one hand, the advantage of its flexibility, a major benefit of this would be the potential use of non-parametric tests. On the other hand, the amount or resources required for large population sizes combined with heavy sampling could end up being limiting both in memory and computational terms.

The alternative chosen for this approach is keeping only the aggregate information that is required to use the Welch test. This is achieved extending the attributes of the individuals so that, in addition to variables that encode the solution, they include summary statistics like the present average and current number of samples the average is based on. The information remains in the system for as long as the individual is in the population.

This offers advantages in both mentioned dimensions. First of all, the amount of memory required is low and fixed as, regardless of the number of samples, the information

carried by individuals is always the same. Then, there is also additional efficiency coming from the fact that the values of all the aggregate statistics required for the test can be updated on-line. This limits considerably the cost of considering each additional sample and allows the test to be performed in a faster manner. The relevance of this feature is directly related to n. The more samples are required by static resampling, the larger the potential benefits would be.

4. Experimental Analysis

Experimentation was performed in order to determine if the proposed efficient sampling achieves a significant reduction of computational costs.

Given the adaptable nature of the strategy, the process starts with the selection of a dominance-based optimization algorithm. NSGA-II and SPEA2 where selected as MOOAs, as they are two of the most widely used techniques in the literature. For these tests both the standard static resampling version and the efficient one were coded on jMetal (Durillo, Nebro, and Alba 2010). This multi-objective optimization and metaheuristics framework allows a better comparison of the alternatives as all the basic components are consistent.

The results of both algorithms, standard and efficient, were tested on extended versions of the ZDT1 to ZDT4 and ZDT6 functions. This popular set of test functions for multiobjective optimizers introduced by Zitzler, Deb, and Thiele (2000) accounts for several problems of a different nature (convex, discrete, multimodal...). Since the original version of these problems does not include a component that requires resampling, an extra objective to be minimized was added, the dispersion indicator described below. The ZDT functions are defined for an arbitrary number of dimensions; 30 dimensions were used for all the functions, which is an important increase in complexity over the value of 10 used for ZDT4 and ZDT6 by Deb et al. (2002).

For the purposes of this work, the robustness of a solution is modelled through an explicit dispersion indicator that can be adapted both to track unstable areas around solutions and to handle noisy fitness functions (García et al. 2014). Having said that, the general approach would also be compatible with other variance definitions based on resampling like the ones described in Gaspar-Cunha and Covas (2008). In this case the control of solution implementation risk and the evaluation of the sensitivity of solutions to direct perturbation, was made through an additional robustness objective that was modeled as follows.

Given a solution $\boldsymbol{x}$, its dispersion is defined as the average distance in the solution space to a set of n samples, $\boldsymbol{x}'_i$, obtained by multiplying the decision variables times a random perturbation. A larger value would indicate that the solution is less robust and vice-versa. Specifically, the perturbation is defined as a vector of uniformly distributed random variables $\boldsymbol{\varepsilon}_i$ whose values are in the range $[1-p, 1+p]$. Here, p specifies the maximum perturbation percentage to be applied on each of the values of $\boldsymbol{\varepsilon}_i$. The robustness R_s of the solution $\boldsymbol{x}$ is defined in 7 as

$$R_{\boldsymbol{x}} = \frac{\sum_{i=1}^{n} d\left(f\left(\boldsymbol{x}\right), f\left(\boldsymbol{x}'_i\right)\right)}{n} \tag{7}$$

where n is the number of similar solutions sampled around $\boldsymbol{x}$ and $d\left(f\left(\boldsymbol{x}\right), f\left(\boldsymbol{x}'_i\right)\right)$ is the distance in the objective space once. The distance mentioned in the previous formula is the Mahalanobis distance (Mahalanobis, P. C. 1936), whose use has the advantage of limiting the problems of scale and correlation that often affect the Euclidean distance.

It is defined as

$$d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sqrt{(\boldsymbol{x}_1 - \boldsymbol{x}_2)^T \sum^{-1} (\boldsymbol{x}_1 - \boldsymbol{x}_2)} \tag{8}$$

where $\sum^{-1}$ is inverse of the variance-covariance matrix.

The previous process can be seen as sampling a multi-dimensional box centred at the solution (variable space) and computing a dispersion metric based on the location of the samples on the objective space. As it was discussed before, this specific dispersion indicator is only one of many potential possibilities and it could be replaced with other alternatives based on the resampling concept.

The experimentation is aimed at studying whether or not the efficient version of the algorithm can offer similar results with fewer fitness evaluations. Such similarity will be measured in terms of the Hypervolume of the fronts as performance indicator (Zitzler and Thiele 1999). Hypervolume calculations require normalization of the values for each objective function. For this purpose, all the Pareto fronts obtained throughout the experiments have been combined in a "reference" front for each of the ZDT functions. The crowding operator was used to select the best 1000 non-dominated solutions in each of these combined fronts. The resulting set of solutions was then considered the "optimal" solution for each of the ZDT problems.

4.1 *Selection of parameters*

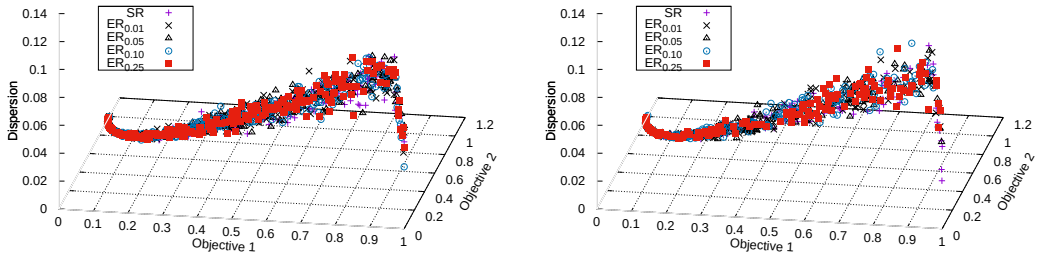
Some preliminary experimentation was performed in order to select a maximum number of samples n that was adequate. This initial effort showed that a small value for n ($n = 5$) resulted in robustness estimates with a rather large variance. This problem was overcome rising the samples to $n = 50$. In a real-world application, proper values of this parameter will depend on the sensitivity of the solution to noise.

Another important parameter that closely depends on the problem being considered is the nature of the perturbation for each of the input variables. A uniform noise distribution that introduces a maximum distortion of 5% simultaneously and independently on each of the input variables was used for the experiments. This parameter is required by the dispersion indicator chosen as definition of robustness, but others might require higher or lower values. Having said that, it is independent of the efficient resampling strategy described in the previous section.

Finally, thresholds for the significance tests that are performed during rank comparison were specified. This is the only parameter, α, that is specific to the component in charge of reducing the number of fitness evaluations, and was arbitrarily set to: 0.01, 0.05, 0.1 and 0.25.

For the NSGA-II algorithm, the typical configuration found in (Deb et al. 2002) for real number representation was used: crossover was performed using Simulated Binary Crossover with crossover probability of 0.9, and mutation was Polynomial Mutation with probability equal to $1/dimensions$; both crossover and mutation used distribution indexes of 20.0. Selection was performed by Binary Tournament.

In order to compare different executions of the algorithm it was important to ensure that the solutions obtained on each run were effectively reaching the optimal Pareto front. This was a concern specially for the more complex ZDT4 and ZDT6 problems. Preliminary experimentation showed that, for 30 dimensions, a population size of 1000 individuals and 1000 generations provided good results. Those values were therefore used in all the experiments. As these algorithms are stochastic, 25 independent runs



(a) NSGA-II (b) SPEA2

Figure 3. ZDT1. Comparison of Pareto fronts obtained by the standard algorithm and efficient resampling for different values of α.

were performed for each combination of problem and method.

The parametrization used for SPEA2 was very similar. The only differences were the increase in the number of generations to 1500, as it took a bit longer to converge in some scenarios, and the need to specify the size of the archive, which was set to 1000.

4.2 *Results*

This section reports the results of the experimental process described above. For each test function, data regarding two key quality indicators: hypervolume of the solution and the number of evaluations required, are provided. Results are shown in the form of tables and figures.

Figures are related to each of the test functions and help to visualize the average difference between the basic setup (standard) and the efficient version for different significance thresholds, α. Each table shows the main descriptive statistics: average, median, minimum and maximum hypervolume; over the mentioned 25 independent experiment runs, plus the associated variance.

The statistical significance of the difference of means was analyzed formally using a number of standard statistical tests that were applied according to the following protocol. First, the normality of the samples of 25 results was checked using a Lilliefors test. In case normality was rejected, Wilcoxon test was used. Otherwise, homoscedasticity was verified using Levenene's test and, depending on the result, the process ended relying on either a Welch test, or a standard t-test.

The structure of the front is depicted in fig. 3. There it can be seen how all solutions overlap regardless of the core algorithm chosen. This is also observable when the quality indicators is analyzed.

Results of the experiments on test function ZDT1 are shown in table 1. Similar hypervolumes were found for different threshold values of the α. Although there seems to be a cost in terms of hypervolume, it is almost negligible. The largest loss on this benchmark is 0.25%, large enough to be significant at 5%, but still rather low in magnitude. It was also observed a pattern that links low values of α to smaller hypervolumes. For SPEA2, the use of the efficient strategy resulted in a loss of 0.01% gain. The difference is so small that equality could not be rejected at 1%. The hypervolume figures for efficient approach show that the value of α has a very marginal impact on the quality indicator.

Depending on the setup, the number of evaluations required to get to a contrasted solution varies widely. SR experiments stand for the standard static resampling and represent an scenario where telling whether an intermediate solution dominates another

Table 1. Evaluations for ZDT1 for different values of the significance threshold (α)

			Average	Median	Var.	Min	Max.	Av. Diff. %
NSGA-II	Hypervol.	SR	0.5828	0.5827	$< 10^{-4}$	0.5817	0.5847	NA
		$ER_{0.01}$	0.5819	0.5820	$< 10^{-4}$	0.5803	0.5833	-0.16%
		$ER_{0.05}$	0.5818	0.5815	$< 10^{-4}$	0.5807	0.5843	-0.18%
		$ER_{0.10}$	0.5814	0.5813	$< 10^{-4}$	0.5798	0.5828	-0.25%
		$ER_{0.25}$	0.5814	0.5813	$< 10^{-4}$	0.5804	0.5833	-0.25%
	Evaluations	SR	51051000.0	51051000.0	0.0	51051000.0	51051000.0	NA
		$ER_{0.01}$	46553902.8	46554555.0	250386760.2	46529062.0	46586343.0	-8.81%
		$ER_{0.05}$	42323796.4	42322292.0	369460895.4	42267638.0	42357510.0	-17.10%
		$ER_{0.10}$	37539889.2	37540648.0	676014604.4	37476472.0	37597841.0	-26.47%
		$ER_{0.25}$	20419001.8	20426377.0	1863358956.9	20339851.0	20489701.0	-60.00%
SPEA2	Hypervol.	SR	0.5772	0.5773	$< 10^{-4}$	0.5747	0.5788	NA
		$ER_{0.01}$	0.5777	0.5776	$< 10^{-4}$	0.5759	0.5794	0.08%
		$ER_{0.05}$	0.5777	0.5778	$< 10^{-4}$	0.5752	0.5790	0.08%
		$ER_{0.10}$	0.5776	0.5776	$< 10^{-4}$	0.5756	0.5799	0.06%
		$ER_{0.25}$	0.5771	0.5772	$< 10^{-4}$	0.5752	0.5790	-0.01%
	Evaluations	SR	75051000.0	75051000.0	0.0	75051000.0	75051000.0	NA
		$ER_{0.01}$	69931849.9	69935577.0	256232878.6	69895250.0	69955769.0	-6.82%
		$ER_{0.05}$	65797001.2	65796623.0	969832100.6	65746030.0	65854589.0	-12.33%
		$ER_{0.10}$	61763642.2	61773627.0	2028533760.7	61638516.0	61826027.0	-17.70%
		$ER_{0.25}$	48038747.2	47989475.0	27753236086.2	47770266.0	48387909.0	-35.99%

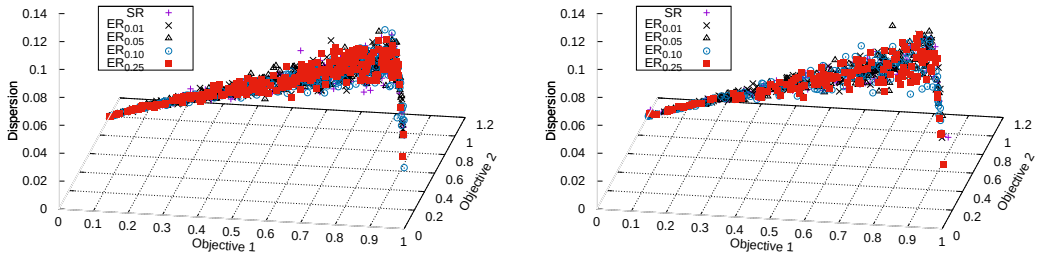
SR: Standard Resampling; ER_α: Efficient resampling

one in terms of robustness requires using all the available information, i.e. robustness is computed by reaching the upper bound limit of n samples. This means that for every evaluated solution 50 samples will be used. Consequently, the efficient resampling approach, ER_α would require fewer evaluations, as the comparison process would be finished as soon as the statistical test rejects the equality.

As expected, the larger the parameter α, the fewer evaluations are required to reject the null hypothesis and, therefore, the more efficient is the strategy. Results in table 1 show that when using a significance level of 25%, i.e. $\alpha = 0.25$, NSGA-II obtains a 60% reduction on the number of evaluations leading to a Pareto front solution whose hypervolume is 0.25% smaller than the one that would have been obtained using all 50 samples in the robustness computations, i.e. static resampling. The general pattern is consistent with the one offered by SPEA2. In this case, even though the advantage in terms of the reduction in the number of evaluations is lower, the gain of 36% is still very sizable. Reducing α makes the rejection more difficult, and consequently, enforces and increase in the number of evaluations. As observed, the most demanding condition within this set of experiments results in 8.81% and 6.82% evaluation gains for NSGA-II and SPEA2 respectively,against a hypervolume reduction of 0.16% and a gain of 0.08%, meaning that the above figures were enough to reject the null hypothesis of equality with at a 1% significance level.

It is possible to conclude this analysis of experiments for ZDT1 saying that even in the worst-case scenarios, offered by NSGA-II, the efficient approach provides significant reductions in the number of evaluations at the cost of very small losses in the hypervolumes of the obtained fronts.

ZDT2 test problem requires optimising a the nonconvex function shown in fig. 4. The pattern displayed in the results for ZDT2 (Table 2) basically mirrors the one observed for ZDT1 for both core algorithms. The hypervolume cost of saving evaluations for NSGA-II is relatively small, below 1%, and the null hypothesis of equality vs. the non-efficient algorithm is rejected at 1%. This, however, is not the case for SPEA2. In this case, equality cannot be rejected at 1% for differences that, in the worst-case scenario, reach 0.16%. The advantage of using the efficient approach with both algorithms is evident.



(a) NSGA-II (b) SPEA2

Figure 4. ZDT2. Comparison of Pareto fronts obtained by the standard algorithm and efficient resampling for different values of α.

Table 2. Evaluations for ZDT2 for different values of the significance threshold (α)

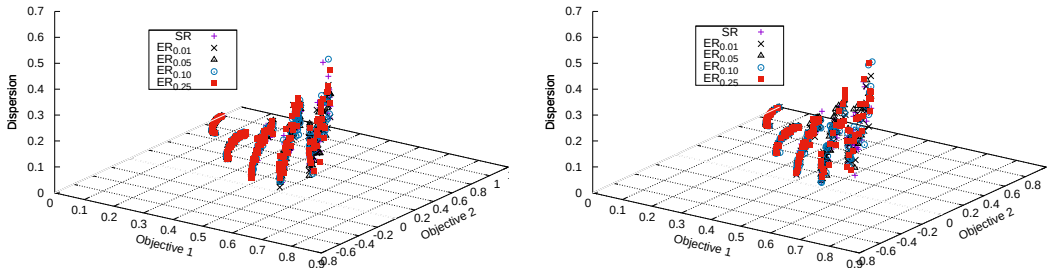
			Average	Median	Var.	Min	Max.	Av. Diff. %
NSGA-II	Hypervol.	SR	0.2404	0.2405	$< 10^{-4}$	0.2384	0.2426	NA
		$ER_{0.01}$	0.2394	0.2395	$< 10^{-4}$	0.2374	0.2411	-0.41%
		$ER_{0.05}$	0.2387	0.2388	$< 10^{-4}$	0.2374	0.2404	-0.71%
		$ER_{0.10}$	0.2388	0.2387	$< 10^{-4}$	0.2368	0.2407	-0.64%
		$ER_{0.25}$	0.2383	0.2383	$< 10^{-4}$	0.2367	0.2400	-0.86%
	Evaluations	SR	51051000.0	51051000.0	0.0	51051000.0	51051000.0	NA
		$ER_{0.01}$	48244286.0	48247062.0	145258140.0	48226908.0	48270627.0	-5.50%
		$ER_{0.05}$	45797333.0	45800434.0	444147145.2	45742774.0	45834670.0	-10.29%
		$ER_{0.10}$	42926288.5	42928124.0	633760289.8	42866870.0	42962938.0	-15.91%
		$ER_{0.25}$	28871642.3	28870287.0	2058248360.4	28768065.0	28959120.0	-43.45%
SPEA2	Hypervol.	SR	0.2333	0.2335	$< 10^{-4}$	0.2318	0.2345	NA
		$ER_{0.01}$	0.2336	0.2334	$< 10^{-4}$	0.2317	0.2361	0.12%
		$ER_{0.05}$	0.2335	0.2335	$< 10^{-4}$	0.2316	0.2360	0.11%
		$ER_{0.10}$	0.2335	0.2331	$< 10^{-4}$	0.2312	0.2359	0.09%
		$ER_{0.25}$	0.2337	0.2337	$< 10^{-4}$	0.2321	0.2364	0.16%
	Evaluations	SR	75051000.0	75051000.0	0.0	75051000.0	75051000.0	NA
		$ER_{0.01}$	71283816.5	71284074.0	257853336.8	71257783.0	71315895.0	-5.02%
		$ER_{0.05}$	68466804.3	68465087.0	871892037.1	68404143.0	68516025.0	-8.77%
		$ER_{0.10}$	65917895.6	65915506.0	1114852965.8	65860730.0	65981257.0	-12.17%
		$ER_{0.25}$	57821886.1	57832002.0	19283501280.3	57561441.0	58124506.0	-22.96%

SR: Standard Resampling; ER_α: Efficient resampling

Table 2 shows that efficient resampling saves between 5.5% and 43.5% of evaluation calls for NSGA-II, or between 5% and 23% if the core algorithm is SPEA2, yet keeping the hypervolumes below the 1% level.

The function ZDT3, graphically described in fig. 5, is much more challenging due to its discrete (non-connected) nature. At first sight, efficient resampling does not seem to result in an evident disadvantage, as the solutions from all the tested configurations seem to cluster in the same places.

This first impression is supported by the data shown in table 3. Average hypervolume for the baseline configuration and the efficient approach are very close regardless of the value of α. They are so similar that, for SPEA2, the equality of differences for the averages vs. standard method cannot be rejected at 1%. For NSGA-II and $\alpha = 0.01$, equality cannot be rejected at 5%. For the rest, it can only be done at 1%. The loss of hypervolume suffered by the robust versions are limited to 0.31% and 0.01%, so the upper bounds are much lower that than for ZDT2. Having said that, the magnitude of the differences is both small and similar.



(a) NSGA-II (b) SPEA2

Figure 5. ZDT3. Comparison of Pareto fronts obtained by the standard algorithm and efficient resampling for different values of α.

Table 3. Evaluations for ZDT3 for different values of the significance threshold (α)

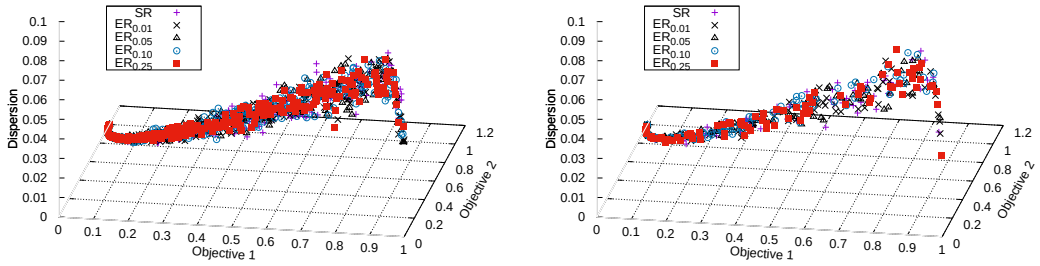
			Average	Median	Var.	Min	Max.	Av. Diff. %
NSGA-II	Hypervol.	SR	0.4956	0.4956	$< 10^{-4}$	0.4936	0.4969	NA
		$ER_{0.01}$	0.4946	0.4946	$< 10^{-4}$	0.4931	0.4964	-0.19%
		$ER_{0.05}$	0.4943	0.4943	$< 10^{-4}$	0.4929	0.4960	-0.26%
		$ER_{0.10}$	0.4945	0.4944	$< 10^{-4}$	0.4934	0.4973	-0.23%
		$ER_{0.25}$	0.4940	0.4941	$< 10^{-4}$	0.4920	0.4955	-0.31%
	Evaluations	SR	51051000.0	51051000.0	0.0	51051000.0	51051000.0	NA
		$ER_{0.01}$	47054253.3	47056512.0	575409700.8	47015692.0	47094058.0	-7.83%
		$ER_{0.05}$	42934043.3	42927974.0	563258007.7	42893429.0	42987260.0	-15.90%
		$ER_{0.10}$	37959222.8	37958697.0	1235802000.9	37871739.0	38028759.0	-25.64%
		$ER_{0.25}$	19167879.7	19170929.0	4309517941.6	19047897.0	19296106.0	-62.45%
SPEA2	Hypervol.	SR	0.4915	0.4915	$< 10^{-4}$	0.4898	0.4933	NA
		$ER_{0.01}$	0.4914	0.4913	$< 10^{-4}$	0.4896	0.4944	-0.01%
		$ER_{0.05}$	0.4916	0.4914	$< 10^{-4}$	0.4899	0.4938	0.04%
		$ER_{0.10}$	0.4915	0.4915	$< 10^{-4}$	0.4898	0.4936	0.02%
		$ER_{0.25}$	0.4915	0.4914	$< 10^{-4}$	0.4901	0.4930	0.01%
	Evaluations	SR	75051000.0	75051000.0	0.0	75051000.0	75051000.0	NA
		$ER_{0.01}$	70109489.8	70100848.0	622444655.7	70065285.0	70151256.0	-6.58%
		$ER_{0.05}$	65948083.6	65943977.0	502876820.0	65902202.0	65994502.0	-12.13%
		$ER_{0.10}$	61541423.2	61540550.0	1683843761.0	61442037.0	61633473.0	-18.00%
		$ER_{0.25}$	46127881.7	46041338.0	120108744344.4	45440490.0	46852811.0	-38.54%

SR: Standard Resampling; ER_α: Efficient resampling

These figures contrast with the efficiency gains. The minimum average number of fitness evaluations saved using NSGA-II is 7.8%, and the maximum is 62.6%, the largest so far, and very close to the results obtained for ZDT1. This pattern is mirrored by SPEA2. Even though the gains are not as important, the reduction of 38.5% is significant and the highest for any of the benchmark functions for SPEA2.

ZDT4 is a function for which the main difficulty to overcome is related to its high multimodality. The structure of the approximations to the Pareto front obtained is shown in fig. 6. We should note again that is difficult to tell any obvious difference between the static and efficient approaches as all the solutions seem to cover a similar area.

The differences in hypervolume are small, and even the parametrization that results in the lowest number of evaluations operates at a expense of 0.27% vs. a gain of 19.15% for NSGA-II in terms of the last column shown in table 4 and an improvement of 14.6% with no hypervolume cost for SPEA2. Despite this, the computational gain is lower than for all other functions, ZDT1-3. This was somewhat expected, as the multimodal nature or the objective functions is likely to create a rugged landscape. This would make



(a) NSGA-II (b) SPEA2

Figure 6. ZDT4. Comparison of Pareto fronts obtained by the standard algorithm and efficient resampling for different values of α.

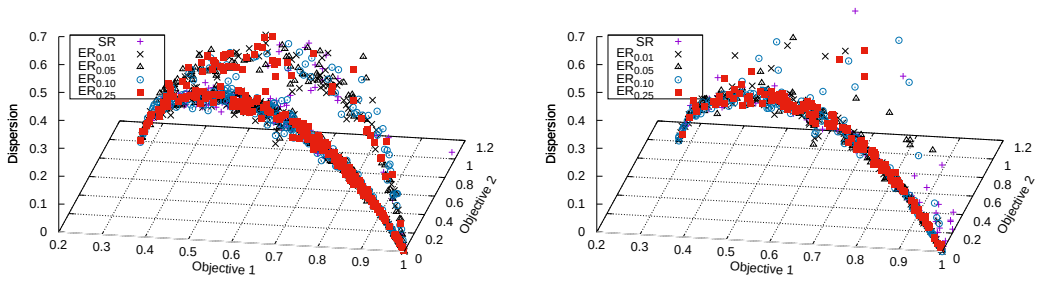
Table 4. Evaluations for ZDT4 for different values of the significance threshold (α)

			Average	Median	Var.	Min	Max.	Av. Diff. %
NSGA-II	Hypervol.	SR	0.5711	0.5711	$< 10^{-4}$	0.5700	0.5730	NA
		$ER_{0.01}$	0.5705	0.5706	$< 10^{-4}$	0.5688	0.5724	-0.11%
		$ER_{0.05}$	0.5703	0.5702	$< 10^{-4}$	0.5688	0.5726	-0.15%
		$ER_{0.10}$	0.5700	0.5701	$< 10^{-4}$	0.5681	0.5714	-0.21%
		$ER_{0.25}$	0.5696	0.5695	$< 10^{-4}$	0.5680	0.5715	-0.27%
	Evaluations	SR	51051000.0	51051000.0	0.0	51051000.0	51051000.0	NA
		$ER_{0.01}$	49106738.2	49107581.0	1144840085.1	49038534.0	49193795.0	-3.81%
		$ER_{0.05}$	47803741.3	47799149.0	2644886587.3	47717531.0	47946096.0	-6.36%
		$ER_{0.10}$	46451380.9	46463460.0	3588566345.1	46313593.0	46568650.0	-9.01%
		$ER_{0.25}$	41275670.8	41286052.0	11534422561.3	41026871.0	41514530.0	-19.15%
SPEA2	Hypervol.	SR	0.5601	0.5601	$< 10^{-4}$	0.5583	0.5629	NA
		$ER_{0.01}$	0.5600	0.5598	$< 10^{-4}$	0.5575	0.5621	-0.02%
		$ER_{0.05}$	0.5603	0.5605	$< 10^{-4}$	0.5584	0.5622	0.04%
		$ER_{0.10}$	0.5598	0.5597	$< 10^{-4}$	0.5583	0.5612	-0.05%
		$ER_{0.25}$	0.5601	0.5600	$< 10^{-4}$	0.5578	0.5633	0.00%
	Evaluations	SR	75051000.0	75051000.0	0.0	75051000.0	75051000.0	NA
		$ER_{0.01}$	71704829.6	71698571.0	2103534014.4	71601328.0	71797216.0	-4.46%
		$ER_{0.05}$	69970344.0	69977941.0	6437428926.6	69842105.0	70145014.0	-6.77%
		$ER_{0.10}$	68399696.0	68405509.0	11608680363.1	68205668.0	68628845.0	-8.86%
		$ER_{0.25}$	64088198.4	64086952.0	21717901773.7	63791446.0	64442298.0	-14.61%

SR: Standard Resampling; ER_α: Efficient resampling

difficult for the statistical test to discriminate clearly due to the variability found in the objective space as it samples the neighborhood of candidate solutions. For this reason, the system requires additional data and there is less margin for efficiency gains. All the cross differences in the number of evaluations are significant at 1%.

To finalize with the set of experiments, the efficient strategy was tested with the non-uniform function ZDT6, as illustrated in figure 7. Results reported in table 5 show that this problem is the hardest of the test set. The reduction in the improvement in the number of evaluations was expected for the same reasons that were mentioned before. However, it is also true that, while the divergences between the standard version and the efficient ones are significant at 1%, we cannot reject equality for the cross differences among the efficient setups with the exception of $\alpha = 0.25$ for NSGA-II (significant at 5%). The loss of hypervolume for NSGA-II is still low, but much larger than for the rest of benchmark functions analyzed. However, the ratio of the loss in one indicator vs. the gain in the other goes from 2.6 to 23.1 for $\alpha = 0.25$. SPEA2 also achieves reductions in



(a) NSGA-II (b) SPEA2

Figure 7. ZDT6. Comparison of Pareto fronts obtained by the standard algorithm and efficient resampling for different values of α.

Table 5. Evaluations for ZDT6 for different values of the significance threshold (α)

			Average	Median	Var.	Min	Max.	Av. Diff. %
NSGA-II	Hypervol.	SR	0.5299	0.5301	$< 10^{-4}$	0.5265	0.5347	NA
		$ER_{0.01}$	0.5266	0.5263	$< 10^{-4}$	0.5240	0.5296	-0.62%
		$ER_{0.05}$	0.5265	0.5262	$< 10^{-4}$	0.5228	0.5312	-0.63%
		$ER_{0.10}$	0.5276	0.5278	$< 10^{-4}$	0.5238	0.5306	-0.43%
		$ER_{0.25}$	0.5270	0.5273	$< 10^{-4}$	0.5217	0.5331	-0.54%
	Evaluations	SR	51051000.0	51051000.0	0.0	51051000.0	51051000.0	NA
		$ER_{0.01}$	50241413.8	50241945.0	385653058.2	50198970.0	50268168.0	-1.59%
		$ER_{0.05}$	49533529.7	49527710.0	1356443116.5	49447424.0	49611966.0	-2.97%
		$ER_{0.10}$	48684614.6	48678209.0	4891676639.9	48560471.0	48882130.0	-4.64%
		$ER_{0.25}$	44674397.1	44686855.0	21229557780.2	44356869.0	44961655.0	-12.49%
SPEA2	Hypervol.	SR	0.5091	0.5093	$< 10^{-4}$	0.5055	0.5132	NA
		$ER_{0.01}$	0.5130	0.5133	$< 10^{-4}$	0.5097	0.5187	0.76%
		$ER_{0.05}$	0.5125	0.5127	$< 10^{-4}$	0.5075	0.5159	0.67%
		$ER_{0.10}$	0.5124	0.5130	$< 10^{-4}$	0.5059	0.5180	0.64%
		$ER_{0.25}$	0.5131	0.5135	$< 10^{-4}$	0.5076	0.5187	0.78%
	Evaluations	SR	75051000.0	75051000.0	0.0	75051000.0	75051000.0	NA
		$ER_{0.01}$	74035031.6	74038535.0	968756660.8	73967752.0	74082528.0	-1.35%
		$ER_{0.05}$	73112445.4	73110770.0	7417017359.9	72953428.0	73290201.0	-2.58%
		$ER_{0.10}$	72057381.3	72049854.0	15931747012.8	71857269.0	72388798.0	-3.99%
		$ER_{0.25}$	68339683.2	68354362.0	41126546984.3	67912340.0	68724942.0	-8.94%

SR: Standard Resampling; ER_α: Efficient resampling

the number of calls to the fitness functions. These are in the range of 1.4% to 8.9% and, as usual, they are more discrete than the ones offered by the other core algorithm.

As it has been shown, for most of the benchmark functions the reductions in the number of evaluations are very significant and they come at the expense of a very small loss in hypervolume. In a real-world application, the decision maker would have to balance the importance of those two and choose a value for α accordingly.

5. Summary and Conclusions

This paper was aimed at introducing a method to increase the computational efficiency of dominance-based multi-objective evolutionary algorithms for robust optimization. The strategy is applicable in problems where robustness is modelled using a dispersion indicator. The approach successfully reduces the number of samples required to compare the robustness of two solutions by means of statistical tests. The implementation allows

on-line updates of the key metrics required to apply them, adding speed and reducing memory needs. The algorithm only depends on a single parameter, namely, the significance level, α, required for the statistical tests.

The strategy was tested for NSGA-II and SPEA2 on a set well-known test functions for multiobjective optimizers (ZDT1-4 and ZDT6), which were extended with a robustness objective. Both the standard algorithms and the modified efficient versions were compared to verify whether or not similar solutions were provided. The results show that the latter required fewer fitness evaluations to provide similar solutions for these functions. This assessment was performed for different values of the parameter α.

The main observation derived from the experimental analysis has been that the efficient implementation resulted in gains in average fitness evaluations than exceeded by a far margin the cost in hypervolume. The magnitudes of these were associated to two main factors, the value of α, and the structure of the fitness landscape.

Generally, the lower the significance level of the statistical test, the closer the results to the baseline implementation would be. For NSGA-II Large α values tended to be associated with slightly smaller hypervolumes. For SPEA2 the evidence is mixed, but most of the differences were so small that they were insignificant at the 1% conventional level. This, however, came together with a a significant reduction of fitness evaluations regardless of the core algorithm. The variations of the hypervolumes of the generated fronts tended to be well below 1% in the worst case scenario while the average gains for the evaluation gains rose up to 62%.

As it was mentioned, the structure of the fitness landscape might limit the performance of the efficient approach. This makes sense as the variance of the dispersion in the objective space of samples around the components of candidate solutions largely influences the ability of the statistical test to reject the null hypothesis of equality. Therefore, in very rugged landscapes, this method would not offer the same advantages in terms of reduction of the number of evaluations.

Researchers and practitioners dealing with optimization problems where resampling is required, and especially on those problems where fitness evaluations are rather expensive, would find the approach described in this paper particularly useful.

Despite the fact that the work presented is based on NSGA-II and SPEA2, the approach is easily applicable to other dominance-based multi-objective algorithms. In addition to this, the idea can also be carried over to other robustness indicators based on dispersion, either new or found in the literature. All this opens room for future research assessing the way that the mentioned components generalize in terms of the quality of the solutions, and the computational effort saved. This could be complemented by studying how the method behaves on real-world applications.

In the field of engineering, the method might be successfully applied in many areas.

In Circuit Design (Patil et al. 2005), for instance, each circuit gate accumulates a total delay based on input-output delays for all the paths involved. These uncertain values define the so-called 'sizing problem' (having minimum delay for a given area or power). Authors suggest that this objective has to be balanced with management of critical paths. So, multi-objective robust optimization might be used to achieve better designs.

Additionally, the method might be of great interest in the field of Structural Design. Robust optimization was used in Ben-Tal and Nemirovski (1997) for truss topology design. Structures are optimized for a series of loading scenarios. The objective of robust design is to ensure that node displacement is admissible even if loads have some variation from the design scenarios. More recently, multi-objective optimization has been considered for structural design problems in Doltsinis and Kang (2004). In this scenario, the approach is very likely to be useful as fitness evaluation are typically costly.

6. Acknowledgements

The authors acknowledge financial support granted by the Spanish Ministry of Economy and Competitivity under grant ENE2014-56126-C2-2-R.

References

Arias-Montano, A., Carlos A. Coello Coello, and E. Mezura Montes. 2012. "Multiobjective Evolutionary Algorithms in Aeronautical and Aerospace Engineering." *Evolutionary Computation, IEEE Transactions on* 16 (5): 662–694.

Ben-Tal, A., and A. Nemirovski. 1997. "Robust Truss Topology Design via Semidefinite Programming." *SIAM Journal on Optimization* 7 (4): 991–1016. http://dx.doi.org/10.1137/S1052623495291951. http://dx.doi.org/10.1137/S1052623495291951.

Beyer, Hans-Georg, and Bernhard Sendhoff. 2007. "Robust optimization - A comprehensive survey." *Computer Methods in Applied Mechanics and Engineering* 196 (33-34): 3190–3218.

Branke, Jürgen. 1998. "Creating Robust Solutions by Means of Evolutionary Algorithms." In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature,* PPSN V. 119–128. London, UK, UK: Springer-Verlag.

Chicano, Francisco, Alejandro Cervantes, Francisco Luna, and Gustavo Recio. 2012. "A Novel Multiobjective Formulation of the Robust Software Project Scheduling Problem." In *Applications of Evolutionary Computation,* Vol. 7248 of *Lecture Notes in Computer Science*497–507. Springer Berlin Heidelberg.

Coello, Carlos A. 2000. "An updated survey of GA-based multiobjective optimization techniques." *ACM Comput. Surv.* 32 (2): 109–143.

Coello, Carlos A. Coello. 1998. "A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques." *Knowledge and Information Systems* 1: 269–308.

Coello Coello, C.A. 2006. "Evolutionary multi-objective optimization: a historical view of the field." *Computational Intelligence Magazine, IEEE* 1 (1): 28 – 36.

Deb, K. 1995. *Optimization for Engineering Design: Algorithms and Examples*. New Delhi: Prentice-Hall.

Deb, Kalyanmoy. 1999. *Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design*. Tech. rep.. Kanpur Genetic Algorithm Laboratory (KanGAL), Indian Institute of Technology, Kanpur.

Deb, Kalyanmoy, and M. Goyal. 1998. "A robust optimization procedure for mechanical component design based on genetic adaptive search." *Transactions of the ASME: Journal of Mechanical Design* 120 (2): 162–164.

Deb, Kalyanmoy, and Himanshu Gupta. 2006. "Introducing robustness in multi-objective optimization." *Evol. Comput.* 14 (4): 463–494.

Deb, Kalyanmoy, and Shivam Gupta. 2010. *Understanding Knee Points and Bicriteria Problems and Their Implications as Preferred Solution Principles*. Tech. rep.. Kanpur Genetic Algorithm Laboratory (KanGAL), Indian Institute of Technology, Kanpur.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. 2002. "A fast and elitist multiobjective genetic algorithm: NSGA-II." *Evolutionary Computation, IEEE Transactions on* 6 (2): 182 –197.

Doltsinis, Ioannis, and Zhan Kang. 2004. "Robust design of structures using optimization methods." *Computer Methods in Applied Mechanics and Engineering* 193 (23): 2221–2237.

Durillo, J.J., A.J. Nebro, and E. Alba. 2010. "The jMetal Framework for Multi-Objective Optimization: Design and Architecture." In *CEC 2010,* Vol. 5467 of *Lecture Notes in Computer Science*4138–4325. Barcelona, Spain: Springer Berlin / Heidelberg. July.

Fonseca, Carlos M., and Peter J. Fleming. 1995. "An overview of evolutionary algorithms in multiobjective optimization." *Evol. Comput.* 3 (1): 1–16.

García, Sandra, David Quintana, Inés M. Galván, and Pedro Isasi. 2012. "Time-stamped Resampling for Robust Evolutionary Portfolio Optimization." *Expert Syst. Appl.* 39 (12): 10722–10730. http://dx.doi.org/10.1016/j.eswa.2012.02.195.

García, Sandra, David Quintana, Inés M. Galván, and Pedro Isasi. 2014. "Extended Mean–variance Model for Reliable Evolutionary Portfolio Optimization." *AI Commun.* 27 (3): 315–324.

Gaspar-Cunha, A., and J.A. Covas. 2008. "Robustness in multi-objective optimization using evolutionary algorithms." *Computational Optimization and Applications* 39 (1): 75–96. http://dx.doi.org/10.1007/s10589-007-9053-9.

Giacobini, Mario, Marco Tomassini, and Leonardo Vanneschi. 2002. "Limiting the Number of Fitness Cases in Genetic Programming Using Statistics." In *Parallel Problem Solving from Nature PPSN VII,* Vol. 2439 of *Lecture Notes in Computer Science* edited by Juan Guervs, Panagiotis Adamidis, Hans-Georg Beyer, Hans-Paul Schwefel, and Jos-Luis Fernndez-Villacaas. 371–380. Springer Berlin / Heidelberg.

Hassan, Ghada, and Christopher D. Clack. 2008. "Multiobjective Robustness for Portfolio Optimization in Volatile Environments." In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation,* Atlanta, GA, USA. GECCO '08. 1507–1514. New York, NY, USA: ACM. http://doi.acm.org/10.1145/1389095.1389387.

Jin, Yaochu, and J. Branke. 2005. "Evolutionary optimization in uncertain environments-a survey." *Evolutionary Computation, IEEE Transactions on* 9 (3): 303 – 317.

Knowles, J., and D. Corne. 1999. "The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation." In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on,* Vol. 1 http://dx.doi.org/10.1109/CEC.1999.781913.

Kruisselbrink, Johannes, Michael Emmerich, and Thomas Bäck. 2010. "An Archive Maintenance Scheme for Finding Robust Solutions." In *Parallel Problem Solving from Nature, PPSN XI,* Vol. 6238 of *Lecture Notes in Computer Science*214–223. Springer Berlin Heidelberg.

Mahalanobis, P. C. 1936. "On the generalised distance in statistics." In *Proceedings National Institute of Science, India,* Vol. 2:149–55. April.

Mann, H. B., and Whitney, D. R. 1947. "On a test of whether one of two random variables is stochastically larger than the other." *Annals of Mathematical Statistics* 18 (1): 50–60.

Marijt, Robert. 2009. "Multi-Objective Robust Optimization Algorithms for Improving Energy Consumption and Thermal Comfort of Buildings." Master's thesis. Faculty of Computer Science, University of Leiden.

Moore, Andrew W., and Mary S. Lee. 1994. "Efficient algorithms for minimizing cross validation error." In *In Proceedings of the Eleventh International Conference on Machine Learning,* 190–198. Morgan Kaufmann.

Moore, J., and R. Chapman. 1999. *Application of Particle Swarm to Multiobjective Optimization.* Tech. rep.. Department of Computer Science and Software Engineering, Auburn University.

Mukhopadhyay, A., U. Maulik, S. Bandyopadhyay, and C.A.C. Coello. 2014a. "Survey of Multiobjective Evolutionary Algorithms for Data Mining: Part II." *Evolutionary Computation, IEEE Transactions on* 18 (1): 20–35.

Mukhopadhyay, A., U. Maulik, S. Bandyopadhyay, and C.A. Coello Coello. 2014b. "A Survey of Multiobjective Evolutionary Algorithms for Data Mining: Part I." *Evolutionary Computation, IEEE Transactions on* 18 (1): 4–19.

Osyczka, Andrzej. 1984. *Multicriterion Optimisation in Engineering.* New York, NY, USA: Halsted Press.

Pareto, Vilfredo. 1896. *Cours D'Economie Politique.* Vol. I and II. F. Rouge, Laussane.

Patil, D., S. Yun, S. J. Kim, A. Cheung, M. Horowitz, and S. Boyd. 2005. "A new method for design of robust digital circuits." In *Quality of Electronic Design, 2005. ISQED 2005. Sixth International Symposium on,* 676–681. March.

Pietro, Anthony Di, Lyndon While, and Luigi Barone. 2004. "Applying Evolutionary Algorithms to Problems with Noisy, Time-consuming Fitness Functions." *Proceedings of IEEE Congress on Evolutionary Computation* 2: 1254–1261.

Ponsich, A., A.L. Jaimes, and C.A.C. Coello. 2013. "A Survey on Multiobjective Evolutionary Algorithms for the Solution of the Portfolio Optimization Problem and Other Finance and Economics Applications." *Evolutionary Computation, IEEE Transactions on* 17 (3): 321–344.

Saha, A., T. Ray, and W. Smith. 2011. "Towards practical evolutionary robust multi-objective optimization." In *Evolutionary Computation (CEC), 2011 IEEE Congress on,* 2123–2130. June.

Sawilowsky, Shlomo S., and R.Clifford Blair. 1992. "A More Realistic Look at the Robustness and

Type II Error Properties of the t Test to Departures From Population Normality." *Psychological Bulletin* 111 (2): 352 – 360.

Siegmund, Florian, Amos H. C. Ng, and Kalyanmoy Deb. 2013. "A comparative study of dynamic resampling strategies for guided Evolutionary Multi-objective Optimization." In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2013, Cancun, Mexico, June 20-23, 2013,* 1826–1835.

Siegmund, Florian, Amos H. C. Ng, and Kalyanmoy Deb. 2015. "Hybrid Dynamic Resampling for Guided Evolutionary Multi-Objective Optimization." In *Evolutionary Multi-Criterion Optimization - 8th International Conference, EMO 2015, Guimarães, Portugal, March 29 -April 1, 2015. Proceedings, Part I,* 366–380.

Syberfeldt, Anna, Amos Ng, Robert I. John, and Philip Moore. 2010. "Evolutionary optimisation of noisy multi-objective problems using confidence-based dynamic resampling." *European Journal of Operational Research* 204 (3): 533–544.

Talbi, El-Ghazali. 2009. *Metaheuristics: From Design to Implementation.* Wiley Publishing.

Tamaki, H., H. Kita, and S. Kobayashi. 1996. "Multi-objective optimization by genetic algorithms: a review." In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on,* 517 –522. May.

Teller, Astro, and David Andre. 1997. "Automatically Choosing the Number of Fitness Cases: The Rational Allocation of Trials." In *Genetic Programming 1997: Proceedings of the Second Annual Conference,* 321–328. Morgan Kaufmann.

Veldhuizen, David A. Van, and Gary B. Lamont. 1998. *Multiobjective Evolutionary Algorithm Research: A History and Analysis.* Tech. rep.. Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology.

Welch, B. L. 1947. "The Generalization of 'Student's' Problem when Several Different Population Variances are Involved." *Biometrika* 34 (1/2): 28–35.

Zitzler, Eckart, Kalyanmoy Deb, and Lothar Thiele. 2000. "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results." *Evol. Comput.* 8 (2): 173–195. http://dx.doi.org/10.1162/106365600568202.

Zitzler, E., M. Laumanns, and L. Thiele. 2001. "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization." In *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems,* edited by K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty. 95–100. Athens, Greece: International Center for Numerical Methods in Engineering.

Zitzler, E., and L. Thiele. 1999. "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach." *Trans. Evol. Comp* 3 (4): 257–271. http://dx.doi.org/10.1109/4235.797969.