Multiobjectivization with NSGA-II on the Noiseless BBOB Testbed

Thanh-Do Tran∗* Dimo Brockhoff†† Bilel Derbel∗∗

∗Inria Lille – Nord Europe, DOLPHIN Team, 59650 Villeneuve d’Ascq, France
†Université Lille 1, LIFL, UMR CNRS 8022, 59655 Villeneuve d’Ascq Cedex, France
firstname.lastname@inria.fr

1. INTRODUCTION

The idea of multiobjectivization, i.e., the reformulation of a single-objective problem by multiple objectives and its resolution by means of a multiobjective optimizer, has been around since the beginning of the new millennium [14, 13]. Two basic ideas can thereby be distinguished: either the single-objective problem is decomposed into two or more objective functions [14] or one or more additional objective functions, so called helper-objectives, are optimized along with the original single-objective function [14, 13]. Several studies report on improving performance for combinatorial problems—early examples range from the traveling salesman problem [14], over reducing bloat in genetic programming [3, 6], to job-shop scheduling [13]. Also for some real-world optimization tasks, multiobjectivization seems to help [9]. The main argument in favor of multiobjectivization for combinatorial problems is thereby the ability to overcome local optima and the possibility of introducing additional search directions on plateaus of equal function values. This is related to the more general idea of increasing population diversity which has been studied independently, see e.g., [18].

Whereas the positive impact of multiobjectivization for combinatorial problems depends highly on the choice of typically problem-dependent objective functions [4, 10], for continuous problems, most studies favor a problem-independent approach, in which the diversity of the algorithm’s population or archive is used as a second objective function [2, 5, 16, 17]. The main argument of [16] to use the distance to the closest neighbor (dcn) in the population of the NSGA-II algorithm [7] as the second objective is that such an objective function “decreases the selection pressure of the original (single-objective) optimization scheme” with the result that “some low-quality individuals could survive in the population with a higher probability” and in turn “these individuals could help to avoid stagnation in local optima” [16].

Unfortunately, in [16, 17], no comparison with other state-of-the-art single-objective methods is performed. Here, we want not only to investigate the impact of multiobjectivization on the performance on the BBOB’2013 noiseless functions [8, 12], but also to see how the approach of [16] compares with state-of-the-art algorithms for numerical optimization. To this end, we used the original implementation of NSGA-II [1] with almost the same algorithmic components as described in [16]. More precisely, we used no additional termination criterion other than the maximum number of function evaluations, performed no restarts, and used the suggested uniform mutation operator as in [16]. In order to investigate the impact of multiobjectivization, we considered an NSGA-II variant with dcn as the second objective (U-dcn) and another where the second objective function was simply set to zero (U-zero). In order to have a better idea of how much the choice of mutation operators affects the search performance, we further compared with the variants...
where NSGA-II’s original polynomial mutation \([7]\) replaces the uniform mutation (denoted by P-\(\text{dcn}\) and P-zero).

More details on the algorithms are given in the next section while Sec. 3 details the experimental procedure. Section 4 presents the mandatory timing experiments and the comparison results in Sec. 5 conclude the paper.

2. ALGORITHM PRESENTATION

2.1 The Artificial Second Objective

There are several ways to introduce artificial objectives into a mono-objective problem, which are in general to be considered as functions measuring the diversity of a population of solutions. In \([17]\), the authors studied the performance of three such functions, namely \(\text{dcn}\) (distance to the closest neighbor of the population), \(\text{adi}\) (average distance to all individuals), and \(\text{dbi}\) (distance to the best individual). They showed that multiobjectivization with \(\text{dcn}\) as a second objective leads to superior performance.

In this study, we shall also use \(\text{dcn}\) for multiobjectivizing the BBOB functions. Having a set of individuals, the \(\text{dcn}\) with respect to individual \(i\) is defined as the Euclidean distance to the closest member of the population, considering that the decision space is real-valued. More formally,

\[
d\text{cn}(i) = \min_j \left\{ \left( \sum_{\ell} (x_{i\ell} - x_{j\ell})^2 \right)^{\frac{1}{2}} \right\},
\]

where \(x_{i\ell}\) is the \(\ell\)-th decision variable w.r.t. individual \(i\).

2.2 The Multiobjective Algorithm

Among a multitude of multiobjective algorithms, we consider the well-known NSGA-II \([7]\) which was also employed in \([17]\). For the sake of reproducibility, we recall the main components of NSGA-II and the way they were implemented in our experiments. First, since the standard NSGA-II deals with minimizing objectives while \(\text{dcn}\) is to be maximized (i.e. to increase diversity), we set the second objective used in NSGA-II to be the maximum \(\text{dcn}\) over all individuals minus the \(\text{dcn}\) of the considered individual. The population size \(N\) was set to be 8. In fact, a small population size was shown to perform relatively well in \([17]\), and the population size in the standard NSGA-II implementation \([1]\) should be a multiple of 4. We used the simulated crossover (SBX) with a distribution index of 15, where each gene (variable) was crossed with a probability of 0.5. As for the mutation, we considered two operators: (i) the uniform mutation (U), and (ii) the polynomial mutation (P) with distribution index \(\eta = 100\). Notice that only the uniform mutation was considered in \([17]\). We set the crossover probability to 1 and the mutation probability to \(\frac{1}{2\cdot D}\), where \(D\) is the number of variables (i.e. problem dimension). While the uniform mutation naturally restricts the variables to an interval (here chosen as \([-5, 5]\)), we also restricted the decision variables to this interval for the polynomial mutation by assigning all the mass of the probability distribution that is outside a variable’s bound to the boundary value.

3. EXPERIMENTAL PROCEDURE

In order to study the impact of multiobjectivization, we considered running NSGA-II while artificially setting the second objective to zero, i.e., all individuals have equal values in their second objectives. This has the effect of turning off the crowding-distance-based selection mechanism specific to NSGA-II, and favoring the selection of individuals having better fitness in the original first objective.

We ended up with four algorithm variants depending on whether \(\text{dcn}\) was switched on or off, and which mutation (U or P) was used. In the remainder, these variants are respectively denoted by U-\(\text{dcn}\), U-zero, P-\(\text{dcn}\), and P-zero.

For our experimentation, we used the standard C implementation of NSGA-II available for free download at \([1]\), and set objectives and parameters to fit in our settings. Moreover, the initial population in NSGA-II was uniformly sampled in \([-5, 5]^D\). We run the four NSGA-II variants up to a budget of \(10^8\) function evaluations or until the maximal BBOB precision of \(10^{-8}\) was reached. It is to notice that there was no independent restart in our implementations. We considered dimensions \(D \in \{2, 3, 5, 10, 20\}\) and all the 15 instances of the BBOB’2013 testbed.

4. TIMING EXPERIMENTS

In order to assess the dependency of the four algorithm variants on the problem dimension, the requested BBOB timing experiments were performed on a Dell XPS 720 machine using the Intel® Core™2 Quad Processor Q6600 running at 2.40 GHz with 2.0 GB RAM. Note that each implementation was deployed exclusively on a single core of the CPU. All implementations were built using the GCC 4.7.2 compiler and executed under the Ubuntu 12.10 Linux distribution. Each algorithm variant was run iteratively on the first instance of \(f_8\) within \(10^7\) function evaluations until at least 30 seconds had passed. This procedure was repeated over seven problem dimensions, i.e. \(D = \{2, 3, 5, 10, 20, 40, 80\}\).

The approximate per-function-evaluation runtimes for U-zero were 7.3, 7.8, 8.5, 10, 13, 18, and 29 times \(10^{-7}\) seconds; for U-\(\text{dcn}\) 12, 13, 16, 23, 38, 90, and 110 times \(10^{-7}\) seconds; for P-zero 9.9, 10, 11, 13, 16, 23, and 36 times \(10^{-7}\) seconds; and for P-\(\text{dcn}\) 13, 16, 19, 28, 42, 58, and 82 times \(10^{-7}\) seconds in 2, 3, 5, 10, 20, 40, and 80 dimensions respectively.

5. RESULTS

Results from experiments according to \([11]\) on the benchmark functions given in \([8, 12]\) are presented in Figures 1, 2, and 3 and in Tables 1 and 2. The expected running time (ERT) used in the figures and tables depends on a given target function value, \(f_t = f_{\text{init}} + \Delta f_t\), and is computed over all relevant trials as the number of function evaluations executed during each trial while the best function value did not reach \(f_t\), summed over all trials and divided by the number of trials that actually reached \(f_t\). \([11, 15]\), Statistical significance is tested with the rank-sum test for a given target \(\Delta f_t\) \((10^{-8}\) as in Figure 1) using, for each trial, either the number of needed function evaluations to reach \(\Delta f_t\) (inverted and multiplied by \(-1\)), or, if the target was not reached, the best \(\Delta f\)-value achieved, measured only up to the smallest number of overall function evaluations for any unsuccessful trial under consideration.

From our experimental results, three main observations are formulated in the following paragraphs. All results mentioned are statistically significant. When comparing two algorithm variants, the significance is thereby checked with the two-algorithm facilities of BBOB (plots not shown here).

Impact of Multiobjectivization: When comparing the \(\text{dcn}\) variants to the variants without \(\text{dcn}\), multiobjectiviza-
tion seems to help on some functions whereas a negative impact can only be observed for a few. For NSGA-II with uniform mutation, U-dcn outperforms U-zero for the separable functions, original Rosenbrock ($f_8$), and the discus function ($f_{11}$ in 20-D) for almost all targets and on $f_{14}$ and $f_{22}$ in 5-D and for the most difficult targets. U-zero is, on the other hand, only better on $f_{14}$ and the separable functions (in 20-D) for easy targets. For NSGA-II with polynomial mutation, the impact of multiobjectivization is less pronounced with a similar tendency on the separable and moderate functions with P-dcn being better only on $f_2, f_6, f_8$, and $f_{14}$. On the sphere ($f_1$) and the linear function $f_5$, on the other hand, the version without the dcn objective is clearly better. The performance differences are larger in higher dimensions.

Impact of Mutation: On some functions, the choice of mutation operator seems to have a stronger impact on the performance than whether multiobjectivization is employed or not. Specifically, when comparing U-dcn with P-dcn, the polynomial mutation gives better results on functions $f_1, f_2, f_5, f_6, f_{11}$, and $f_{14}$ for the most difficult targets while the uniform mutation is typically better at the beginning of the search. This behavior is not surprising as the distribution index $\eta = 100$ for the polynomial mutation was fixed throughout the search. As a result, the mutation's step size is typically too small at the beginning of the search but better suited at later stages. On $f_{20}, f_{21}$, and $f_{22}$, the uniform mutation is, however, interestingly better for all targets.

Competitiveness: The third observation is that the ERTs of all the four variants are still far from being competitive with the artificial best algorithm from BBOB’2009.

It is worth noticing that, initially, our goal was not to design an optimizer that would perform competitively compared to existing state-of-the-art single-objective algorithms, which normally use advanced optimization techniques like step size adaptation. Notice also that we have conducted some other preliminary experiments using another advanced multiobjective algorithm, namely R2-EMOA, and we observed better ERTs, but a seemingly comparable impact of dcn. This suggests that dcn or any other alternative objectives may, to some extent, be beneficial if carefully combined with an appropriate multiobjective algorithm or variation operator (e.g. a specific mutation). From our experiments, however, we can only conclude that using dcn with the specified NSGA-II is showing a limited potential for tackling the noiseless single-objective BBOB testbed.

Acknowledgements

The authors acknowledge the support of the French national research agency (ANR) within the Modèles Numérique project “NumBBO - Analysis, Improvement and Evaluation of Numerical Blackbox Optimizers” (ANR-12-MONU-0009-03).

6. REFERENCES


Figure 1: Expected running time (ERT in number of f-evaluations) divided by dimension for target function value $10^{-8}$ as $\log_{10}$ values versus dimension. Different symbols correspond to different algorithms given in the legend of $f_1$ and $f_{24}$. Light symbols give the maximum number of function evaluations from the longest trial divided by dimension. Horizontal lines give linear scaling, slanted dotted lines give quadratic scaling. Black stars indicate statistically better results compared to all other algorithms with $p < 0.01$ and Bonferroni correction number of dimensions (six). For $f_{10}$, $f_{15}$, $f_{16}$, and $f_{23}$, no finite ERT could be displayed (except for 2-D), therefore the empirical cumulative distribution graphs per function (similar to Figs. 2 and 3) are shown instead (5-D: dashed lines; 20-D: straight lines). Legend: ○:P-zero, ▽:P-DCN, ⋆:U-zero, ◻:U-DCN.
Figure 2: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/D) for 50 targets in $10^{[-8..2]}$ for all functions and subgroups in 5-D. The “best 2009” line corresponds to the best ERT observed during BBOB 2009 for each single target.
Figure 3: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/D) for 50 targets in $10^{[-8..2]}$ for all functions and subgroups in 20-D. The “best 2009” line corresponds to the best ERT observed during BBOB 2009 for each single target.
| $\Delta f_{opt}$ | 1e1 | 1e0 | 1e-1 | 1e-3 | 1e-5 | 1e-7 | $\#\text{succ}$ | $\Delta f_{opt}$ | 1e1 | 1e0 | 1e-1 | 1e-3 | 1e-5 | 1e-7 | $\#\text{succ}$ | $\Delta f_{opt}$ | 1e1 | 1e0 | 1e-1 | 1e-3 | 1e-5 | 1e-7 | $\#\text{succ}$ |
| 4.0 | 15/15 | 15/15 | 0/15 | 11/15 | 83 | 87 | 88 | 90 | 92 | 94 | **| | | | | | | | | | | | | | | |
| 3.2 | 0/15 | 0/15 | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4.3(4) | 4.7e4(6e4) | 7.8e4(9e4) | | | | | | | | | | | | | | | | | | | | | | | | | |
| **| 1.1(0.7) | 6.1(4) | 2.5(3) | 447(538) | 2846(3800) | 9233(8613) | | | | | | | | | | | | | | | | | | | |
| 2.3e4(2e4) | 1.4e5(2e5) | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5.6(8) | 3.3e4(4e4) | 2.3e4(3e4) | 25(18) | 240(136) | 988(587) | 3.7e4(8e4) | 2.6e5(4e5) | 1.5e6(2e6) | | | | | | | | | | | | | | | | | | | |
| 2.6e4(4e4) | 3.6e4(5e4) | 2.1e4(3e4) | 2.0e4(3e4) | 1.9e4(2e4) | 1.9e4(3e4) | | | | | | | | | | | | | | | | | | | |
| 7.3(7) | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| The measured during BBOB-2009 (given in the respective first row) for different $f_{opt}$ values in dimension 5. The central 80% range divided by two is given in braces. The median number of conducted function evaluations is additionally given in italics, if $ERT(10^{-7}) = \infty$. $\#\text{succ}$ is the number of trials that reached the final target $f_{opt} + 10^{-8}$. Best results are printed in bold. Algorithm names are abbreviated (e.g. P-0 is P-zero).
Table 2: Expected running time (ERT in number of function evaluations) divided by the respective best ERT measured during BBOB-2009 (given in the respective first row) for different \( \Delta f \) values in dimension 20. The central 80% range divided by two is given in italics. The median number of conducted function evaluations is additionally given in italics, if \( \text{ERT}(10^{-2}) = \infty \). \#succ is the number of trials that reached the final target \( f_{\text{opt}} + 10^{-8} \). Best results are printed in bold. Algorithm names are abbreviated (e.g. P-0 is P-zero).