Performance Assessment of Multi-Objective Evolutionary Algorithms With the R Package ecr

Jakob Bossek
University of Münster
Münster, Germany
bossek@uni-muenster.de

ABSTRACT

Assessing the performance of stochastic optimization algorithms in the field of multi-objective optimization is of utmost importance. Besides the visual comparison of the obtained approximation sets, more sophisticated methods have been proposed in the last decade, e. g., a variety of quantitative performance indicators or statistical tests. In this paper, we present tools implemented in the R package `ecr`, which assist in performing comprehensive and sound comparison and evaluation of multi-objective evolutionary algorithms following recommendations from the literature.

CCS CONCEPTS

• **Software and its engineering** → **Frameworks**; • **Computing methodologies** → *Search methodologies*;

KEYWORDS

Software-Tools, Evolutionary Optimization, Performance Assessment

ACM Reference Format:
Jakob Bossek. 2018. Performance Assessment of Multi-Objective Evolutionary Algorithms With the R Package ecr. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan,* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3205651.3208312

1 INTRODUCTION

In many application domains, e. g., logistics or machine scheduling, the simultaneous optimization of multiple, usually conflicting, objectives is required [22]. It turned out that in the field of *multi-objective optimization* (MOO) stochastic optimization algorithms such as evolutionary multi-objective algorithms (EMOA) and ant-colony-algorithms (ACO) often perform extraordinarily well [11], distinguishing themselves by high robustness and applicability even under difficult circumstances, e. g., in a black-box scenario with little or no knowledge of the underlying objective functions. A plethora of algorithmic approaches and modifications have been proposed in the past decades (e. g., [2, 12]) and with all these proposals the requirement for sophisticated and statistically sound methods for performance evaluation and comparison of randomized search heuristics arose. Visual comparison of the obtained Pareto-front approximations is a good starting point and occasionally already reveals some insights, but a rigorous performance analysis requires more than that. Often, the statistical programming language R is adopted for this purpose due to its status of the "lingua france" of data science. However, until recently there was no open-source collection of functions for comprehensive EMOA performance assessment in R. In this paper, we give a tutorial on the EMOA performance assessment module which was shipped with the recent version of the R package `ecr`. The provided methods and workflows are inspired by the tutorial by Knowles et al. [16] on performance assessment of stochastic multi-objective optimizers. Partially, we interface the accompanying tools, which are available in the Platform and Programming Language Interface PISA [5], directly. The module relies on few functions which facilitate the most common tasks, e. g., scatterplots of approximation sets, computation and visualization of performance indicators and statistical hypothesis tests beside others. The modules are highly extensible, e. g., by custom performance indicators, and thus a valuable toolset for researchers and practicioners working the the field of multi-objective (evolutionary) optimization.

The remainder of the paper is organized as follows: Section 2 introduces some notation and gives a brief overview of performance assessment in multi-objective (evolutionary) optimization. Next, in Section 3 some background information on `ecr` is provdided together with some requirements of the EMOA performance assessment module. A hands-on tutorial on the latter is given in Section 4 before we conclude the paper and give an outlook on future work in Section 5.

2 MULTI-OBJECTIVE OPTIMIZATION

Here we give a very brief introduction into multi-objective optimization and performance measurement establishing a foundation/vocabulary for the tutorial in Section 4. We leave aside details wherever possible. For a thorough introduction to multi-objective optimization we refer to [11]. For profound work on performance assessment we recommend, e. g., [15, 16].

In multi-objective optimization we are given a vector-valued function $F : S \to D$, where S is the *decision space* (either numeric or discrete or mixed) and D is the p-dimensional *objective space* (most often $D \subseteq \mathbb{R}^p$) with $p \geq 2$ usually conflicting objectives. W. l. o. g. all objectives are to be minimized. Since there is no total order in D, the notion of optimality is different to the one in single-objective optimization. Given two solutions $x, y \in S$ we say that x *dominates*

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07…$15.00
https://doi.org/10.1145/3205651.3208312

y, denoted as $x \preceq y$ if the corresponding objective vectors behave in the same way, i. e., if $F(x) \preceq F(y)$. Here, the binary *dominance relation* is defined as follows: $x \prec y :\Leftrightarrow F_i(x) \leq F_i(y)\ \forall i = 1, \ldots, n$ and $\exists j \in \{1, \ldots, n\} : F_j(x) < F_j(y)$. Fig. 1 shows an example. Here A dominates C and D, but A and B are incomparable. The goal for an multi-objective optimizer is to approximate the set of non-dominated solutions, i. e., the set $\text{ND}(S) = \{x \in S \mid !\exists y \in S : F(y) \preceq F(x)\}$ of mutually non-dominated elements termed the *Pareto-set* and/or its image $F(\text{ND}(S))$ in objective space, termed the *Pareto-front* (the circles in Fig. 1).

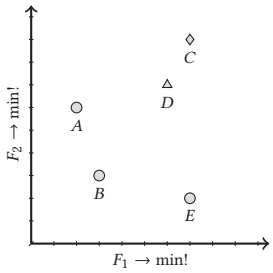

Figure 1: Illustration of Pareto-dominance concept.

Bio-inspired stochastic algorithms like evolutionary algorithms have proven to be well-suited to tackle this task in various studies. However, given approximation sets, i. e., approximations to the Pareto-set/front, of multiple runs of several stochastic algorithms the researcher/practicioner is usually confronted with the challenging task of deciding which algorithm performs best on a given problem or set of problems. Here the tradeoff complexity of the multi-objective optimization problem naturally carries over to the performance assessment. Visual comparison of the approximation sets may be a vague indicator. However, this method runs into its limits at the latest if $p > 3$. In the last decades, researchers came up with systematic means for performance assessment of multi-objective stochastic optimizers. Here, we focus on the best-practice recommendations given in the the excellent paper by Knowles et al. [16]. Basically, the authors outline three comparison methodologies, which are all based on aggregating approximations sets into a metric space with an underlying total order: 1) a dominance ranking approach, 2) an indicator-based comparison and 3) attainment functions. Here, we briefly describe the former two as they are used for illustration in Section 4.

Dominance Ranking. Here, Knowles et al. [16] propose to assign each approximation set $C_i \in C$ (C being the union of all approximation sets) for a particular problem instance a rank

$$\text{rank}(C_i) = 1 + |\{C_j \in C : C_j \prec C_i\}|,$$

where $\prec$ is the "better" preference relation on approximation sets: $C_i \prec C_j$, if each $y \in C_j$ is dominated by at least one $x \in C_i$. Thus, each approximation set is reduced to a single integer value. After ranking, the rank distributions can be visually inspected or statistical rank tests may be applied in order to come up with a first assessment.

Quality indicators. A (unary) quality indicator $I : \Omega \rightarrow \mathbb{R}$ maps the space of approximation sets Ω to the real numbers and usually allows for some kind of preference integration.

A simple unary indicator is the *Overall Nondominated Vector Generation* (ONVG, [15]). It is an easy to compute cardinal measure that assigns each approximation set $C \in \Omega$ the number of distinct nondominated points, i. e., $I_O(C) = |\text{ND}(C)|$. However, since a single point may dominate each point from a set it is not difficult to come up with an example where $I_O(C_1) \gg I_O(C_2)$, but C_2 is clearly better. Hence, ONVG should never be used exclusively.

One of the most frequently used quality indicators is the *hypervolume indicator* I_{HV} (also known as S-metric) proposed by Zitzler and Thiele [26]. It measures the amount of space weakly dominated by the approximation set with respect to an anti-optimal reference point R[1] (see Fig. 2 left). Even though studies revealed weaknesses of unary indicators (see, e. g., [27]), in particular the hypervolume indicator remains a popular choice due to some desirable properties, e. g., Pareto-compliance with the dominance relation [15]. Given a reference set $R^* \in \Omega$, e. g., the known true Pareto-front or some approximation of it, the hypervolume difference $I_{\text{HV}}(C) = I_{\text{HV}}(R^*) - I_{\text{HV}}(C), C \in \Omega$ is an alternative measure which is to be minimized (Fig. 2 right).

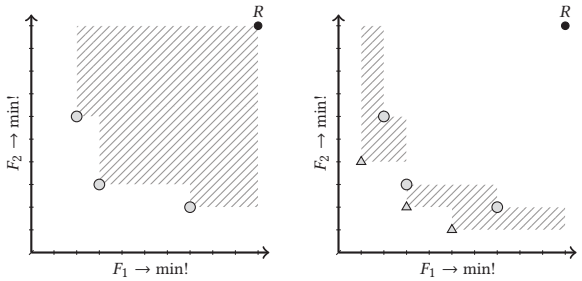

Figure 2: Illustration of dominated hypervolume indicator (left) and hypervolume difference to a reference set (triangles in right hand plot).

Clearly, both methodologies go hand in hand with information loss due to the aggregating nature. On the one hand, this step enables the application of statistical tests and interpretation. On the other hand one should never stick to a single methodology. Instead the recommended workflow is to apply dominance ranking as a first step, take a glance at the approximation sets itself, adopt multiple unary and/or binary quality indicators and apply significance tests whereever possible.

3 EMOA PERFORMANCE ASSESSMENT WITH ECR

The R package `ecr` [6] is a flexible framework for single- and multi-objective optimization. In the last years it was used by researchers to tackle various problems in the field of single-objective combinatorial optimization [10] and multi-objective optimization in particular [6–10, 17]. During the latter studies performed by the

[1]The reference point may be given by explicit bounds of the objective space or an estimation based on the union of all approximation sets

author, a lot of utility functions for EMOA performance assessment had been implemented, which finally found their way into the last major release of the package. After a brief discussion of related R packages, this section gives a gentle introduction into the tools as well as into some requirements and assumptions.

3.1 Related Software

There is a plethora of packages for the R programming language. However, there are few packages dealing with multi-objective optimization: `SPOT` [1], `GPareto` [3], `moko` [20] and `mlrMBO` [4] provide functionality for multi-objective surrogate-assisted optimization, which is of particular interest in the computationally expensive black-box scenario. Multi-objective evolutionary computation frameworks are limited to `mco` [19], `emoa` [18] and `ecr` [6]. While `emoa` contains implementations of some performance indicators, to the best of our knowledge only `ecr` provides broad functionality regarding EMOA performance assessment.

3.2 Overwiew

Figure 3 gives an overview of the performance assessment tools implemented in `ecr`. The data is required to be stored in a data.frame (see Section 3.3 for details), a fairly common data structure in R. As soon as the data is available in the required format the optional data preprocessing may be performed. Normalization of approximation sets to the unit square $[0, 1]^p$ via `ecr::normalize`, filtering interesting problems/algorithms or combinations of both with base R, or the data processing tools of choice (e. g., packages `dplyr` [25] or `data.table` [13]). Moreover, estimation of reference sets or reference points may be neccessary.

Afterwards, performance assessment methods can be applied. Here, `computeDominanceRanking` implements the ranking approach as decribed in Section 2 and `computeIndicators` performs calculation of a set of unary and/or binary indicators. Various arguments may be adapted for customization. E. g., a list of reference points (argument `ref.points`) may be passed to `computeIndicators`. They are passed down to all indicator calculation functions. However, `ecr` sticks to reasonable defaults, if no reference points or sets are passed. In this case(s) reference points are calculated on basis of the union of all approximation sets on a per-problem basis. Analogously, reference sets are approximated as the non-dominated points from the union of all approximation sets in the data. Again, this is done for each problem separately. Results can be visualized with utility functions (e. g., `plotScatter2d`, `plotDistribution`). All plot functions return a flexible `ggplot2` object, which allows for heavy customization (see [24]). To sum up, the performance assessment module of `ecr` has the following (growing) feature list:

- Data preprocessing: filtering, normalization etc.
- Computation of dominance ranking.
- Extensible set of build-in unary and binary performance indicators. At the moment of writing the following indicators are available: hypervolume indicator Zitzler and Thiele [26], ϵ-indicator [27], overall nondominated vector generation / ratio [15], R-indicator family [14] with different utility functions, Minimum distance (MD) [21], (inverted) generational and distance [23].
- Visualization of Pareto-front approximations (2D and 3D), indicator/ranking distributions.
- Statistical hypothises tests.
- Export to LaTeX-tables.

3.3 Required data format

In order to apply `ecr`s performance assessment tools the data collected in experiments must be converted to the required format. Here, we assume that we applied some stochastic multi-objective algorithms on a set of problems each $m > 1$ times resulting in a set of p-tuples (the non-dominated points) for each combination of algorithm, problem, and run. `ecr` requires the data to be stored in a *data.frame*, i. e., basically a tabular representation with columns f1, …, fp containing the objective vectors (each one per line), `prob` (problem name), `algorithm` (name of optimizaton algorithm) and `repl` (number of algorithm run). Table 1 shows an extract from the data set that will serve for demonstration in Section 4. Note that this "long" format comes along with some redundancy. It is, however, quite common in advanced data analysis in R and is required by many efficient data-processing libraries (e. g., `ggplot2` and `dplyr`) used internally by the `ecr` performance assessment methods. This requirement may seem like a strict limitation at first glance. However, further columns with additional meta-data may be added in subsequent steps.

f1	f2	prob	repl	algorithm
835.0	1225.1	instance-040-1	1	SMSEMOA.ZHOU
1917.2	582.6	instance-040-1	1	SMSEMOA.ZHOU
1747.3	584.6	instance-040-1	1	SMSEMOA.ZHOU
1504.1	599.5	instance-040-1	1	SMSEMOA.ZHOU
1626.1	589.7	instance-040-1	1	SMSEMOA.ZHOU
846.7	984.2	instance-040-1	1	SMSEMOA.ZHOU
⋮	⋮	⋮	⋮	⋮
2940.0	1263.3	instance-100-3	10	NSGA2.MIXED
1571.1	1935.2	instance-100-3	10	NSGA2.MIXED
1495.7	2008.1	instance-100-3	10	NSGA2.MIXED

Table 1: Example for data format required.

4 EXAMPLE CASE STUDY

In this section we give a hands-on tutorial on EMOA performance evaluation with `ecr`. The code snippets are wrapped up in a file provided on the official GitHub page https://github.com/jakobbossek/ecr2/blob/master/inst/examples/EMOA.PA.mcMST.R.

4.1 Setting up the workspace

We recommend the most recent official release version (v2.1.0 at the moment of writing) of `ecr` to retrace the subsequent steps. However, to check out the latest (experimental) features, the development version may be installed directly by typing the following code chunk into an interactive R session (note that package `devtools` needs to be installed beforehand).

```
devtools::install_github("jakobbossek/ecr2")
```

Moreover, the packages `dplyr` and `ggplot2` are required. The former interacts smoothly with `ecr` and serves for elegant and efficient

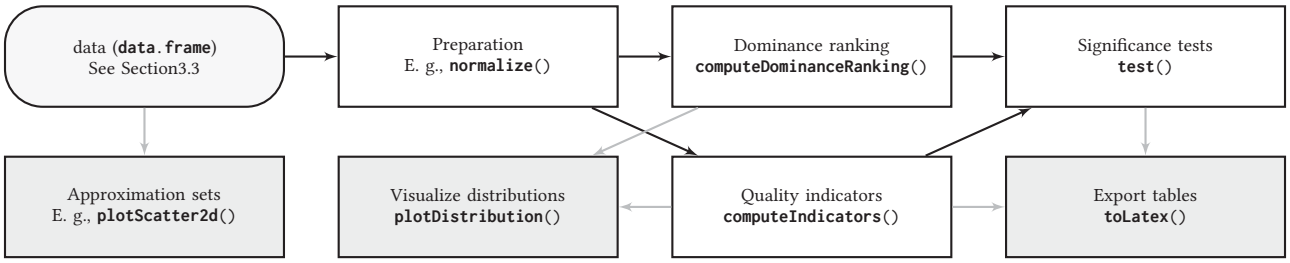

Figure 3: Relation of methodologies/methods from the performance assessment module in ecr inspired by Figure 11 in Knowles et al. [16]. Boxes with white background represents core functionality for performance assessment. Boxes with light gray background contain utilities for data visualization and export.

data manipulation. The latter is a powerful visualization tool used internally by the ecrś plot functions.

```
1 library(ecr)
2 library(dplyr)
3 library(ggplot2)
```

To clarify the package membership of functions used on the following pages, each call to external library functions is prefixed with the package name and the double colon operator. E. g., we write `ecr::normalize` instead of `normalize`.

4.2 Data basis

The data basis is the *mcMST* dataset embedded in ecr for the purpose of demonstration. This data stems from a recent study by Bossek and Grimme [7] on the multi-criteria minimum-spanning-tree problem (mcMST). In a nutshell: The authors introduced a new mutation operator and performed a comparative study with several mutation approaches from the literature. Recombination was neglected to investigate the pure influence of mutation; NSGA-II [12] and SMS-EMOA [2] served as encapsulating wrappers. For more information we refer the reader to [7].

In total the data set contains approximation sets for each 10 independent runs of two meta-heuristics (NSGA-II and SMS-EMOA) with four different mutation operators (SG as proposed by Bossek and Grimme, EX: simple 1-edge-exchange, ZHOU: random swap on Prüfer encoding and MIXED which is a combination of SG and EX) on nine problem instances.

4.3 Performance assessment

First, we import the data set and take a glance at it. The data already fullfils the requirements of the data format (see Section 3) and thus no preprocessing is neccessary, in general. However, for our case study, we focus on NSGA-II results and three instances with 100 nodes each. Therefore we subset the corresponding observations with `dplyr::filter`. Moreover, we normalize the Pareto-front approximations to $[1, 2]^2$.

```
1 data(mcMST)
2
3 obj.cols = c("f1", "f2")
4 mcMST = dplyr::filter(mcMST,
5   grepl("100", prob), grepl("NSGA", algorithm))
6
7 mcMST = ecr::normalize(mcMST,
8   obj.cols = obj.cols, offset = 1)
9
10 head(mcMST)
11 > # A tibble: 6 x 5
12 >       f1    f2 prob              repl algorithm
13 >    <dbl> <dbl> <chr>            <int> <chr>
14 > 1  1.36  1.79 instance-100-1      1 NSGA2.ZHOU
15 > 2  1.73  1.37 instance-100-1      1 NSGA2.ZHOU
16 > 3  1.40  1.66 instance-100-1      1 NSGA2.ZHOU
17 > 4  1.40  1.73 instance-100-1      1 NSGA2.ZHOU
18 > 5  1.67  1.38 instance-100-1      1 NSGA2.ZHOU
19 > 6  1.59  1.41 instance-100-1      1 NSGA2.ZHOU
```

Dominance ranking. Next, we apply the *dominance ranking* approach as proposed by Knowles et al. [16] to get first insights in the performance of the algorithms. The rank distributions are then visualized via boxplots by calling `ecr::plotDistribution`.

```
1 ranks = ecr::computeDominanceRanking(
2   mcMST,
3   obj.cols = obj.cols)
4 ecr::plotDistribution(ranks) +
5   ggplot2::theme(legend.position = "none")
```

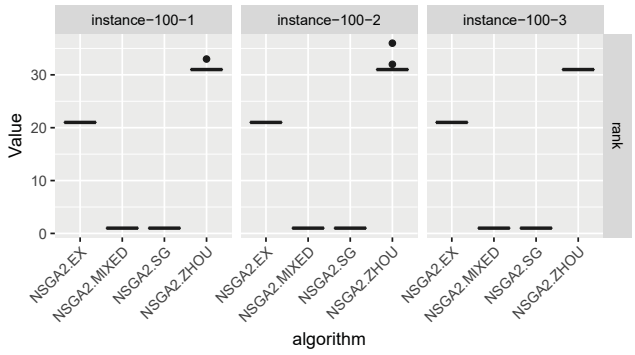

Figure 4: Distribution of dominance ranking.

Note, that the generated `ggplot2` object in line 4 can be customized in any way `ggplot2` allows for. Here, we drop the legend which is included by default for illustration purposes. The output boxplot is shown in Figure 4. Seemingly, MIXED and SG have rank 1 for all instances (and thus produce incomparable approximations); EX and ZHOU are far off with ZHOU being ranked worst.

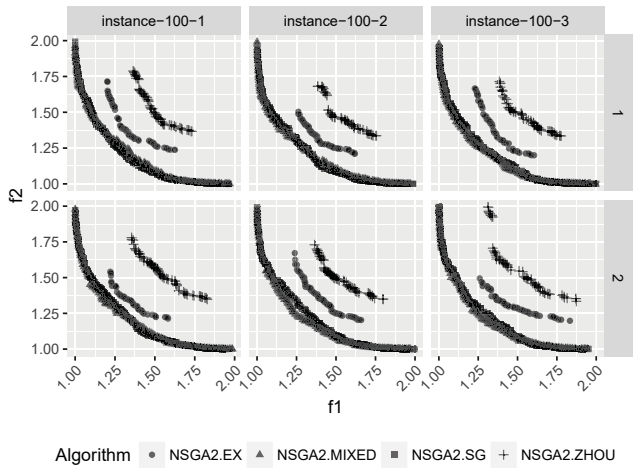

Figure 5: Examplary 2D scatterplots of approximation sets.

Approximation sets. In a next step, we take a look at the non-aggragated approximation sets for each the first two algorithm runs. To do so, we make use of `dplyr::filter` function to select the observations for the runs, and pass the results to `ecr::plotScatter2d`. The passed `facet.*` arguments[2] override the defaults and allow to fine-tune the look of the plot: here we want each one plot for each combination of replication (`repl`) and problem (`prob`) and points should be shaped by `algorithm`. In addition, we adapt the colours to gray-scale by appending a `ggplot` command.

```
ecr::plotScatter2d(
  dplyr::filter(mcMST, repl <= 2),
  facet.type = "grid",
  shape = "algorithm"
  facet.args = list(facets = formula(repl ~ prob))) +
   ggplot2::scale_colour_grey(end = 0.8)
```

The results of the dominance ranking are confirmed. ZHOU always performs worst with its approximation sets being far away from the others. EX is ranked second while, MIXED and SG approximation sets are incomparable. For the latter, observations suggest comparative behaviour close to the lexicographic optima. However, points from the MIXED approximation sets seem to dominate SG approximations in the center. From now on, performance quantification via EMOA performance indicators is required to further evaluate performance differences.

Performance indicators. A common recommendation in literature on multi-objective performance assessment is the usage of multiple performance indicators rather than relying on a single one. Thus, following this best-practise, we decide upon the hypervolume indicator I_{HV}, the unary ϵ-indicator I_ϵ and Overall Nondominated Vector Generation (ONVG) I_O. For demonstration purposes we pass a custom function to compute the latter, while we stick to `ecr`'s implementations of the former two. To do so, we call `ecr::makeEMOAIndicator` and pass the actual function (here simply returns the number of non-dominated points in the approximation set), an internal name, a formula for LaTeX-representation of the indicator and a flag, which indicates the desired direction of higher quality (for ONVG more solutions are preferred).

[2]Details on socalled *facetting* in `ggplot2` is out of scope of this paper. See Wickham [24] for a comprehensive introduction.

Table 2: Table of summary statistics (arithmetic mean) and standard deviation of selected indictors I_{HV} and I_ϵ for all three considered instances. Minimal values are bold on a per-instance basis.

		I_{HV}		I_ϵ	
Problem	**Algorithm**	**Mean**	**StdDev**	**Mean**	**StdDev**
instance-100-1	NSGA2.EX	0.279	0.008	0.226	0.011
	NSGA2.MIXED	**0.020**	0.008	**0.030**	0.010
	NSGA2.SG	0.028	**0.007**	0.037	**0.008**
	NSGA2.ZHOU	0.473	0.019	0.363	0.015
instance-100-2	NSGA2.EX	0.288	0.013	0.233	0.017
	NSGA2.MIXED	**0.016**	**0.004**	**0.031**	**0.006**
	NSGA2.SG	0.031	0.006	0.042	0.006
	NSGA2.ZHOU	0.480	0.018	0.381	0.014
instance-100-3	NSGA2.EX	0.293	0.013	0.227	0.017
	NSGA2.MIXED	**0.022**	**0.006**	**0.037**	**0.008**
	NSGA2.SG	0.033	0.008	0.048	0.011
	NSGA2.ZHOU	0.496	0.010	0.384	0.023

```
myONVG = ecr::makeEMOAIndicator(
  fun = function(points, ...) ncol(points),
  name = "ONVG",
  latex.name = "I_{O}",
  minimize = FALSE
)
```

Now, we set up a list of indicator functions and pass these to the correpsonding `ecr` function `ecr::computeIndicators`, which returns a named list. The results are stored in a data frame and may be easily processed further, e. g., by passing it to `ecr::plotDistribution`. The resulting indicator distributions are depicted in Figure 6. The LaTeX-table generated by line 20 of the subsequent listing is shown in Table 2.

```
unary.inds = list(
  list(fun = ecr::emoaIndHV),
  list(fun = ecr::emoaIndEps),
  list(fun = myONVG)
))

inds = ecr::computeIndicators(
  mcMST, unary.inds = unary.inds
)

head(inds$unary)
>     algorithm           prob repl    HV   EPS ONVG
> 1    NSGA2.EX instance-100-1    1 0.285 0.237   69
> 2 NSGA2.MIXED instance-100-1    1 0.029 0.047  164
> 3    NSGA2.SG instance-100-1    1 0.021 0.033  178
> 4  NSGA2.ZHOU instance-100-1    1 0.491 0.366   71
> 5    NSGA2.EX instance-100-2    1 0.296 0.261   71
> 6 NSGA2.MIXED instance-100-2    1 0.022 0.039  163

ecr::plotDistribution(inds$unary,plot.type = "boxplot")
toLatex(inds$unary, stat.cols = c("HV", "EPS"))
```

Once again, the insufficient performance of ZHOU and EX becomes clear alongside all three indicators. Regarding MIXED and SG, we observe MIXED to perform slightly better than SG regarding both I_{HV} and I_ϵ.In contrast, SG performs better with respect

Table 3: Tables with results of significance tests for all considered instances. There is a square matrix-style subtable (all pairs of algorithms) for each instance and indicators. A special formatter function transforms p-values to scientific notation for better readability and highlights significant results in bold-face.

	instance-100-1 / I_{HV}				**instance-100-1 / I_ϵ**				**instance-100-1 / I_O**			
	EX	MIXED	SG	ZHOU	EX	MIXED	SG	ZHOU	EX	MIXED	SG	ZHOU
EX	-	$\mathbf{5.41 \times 10^{-6}}$	$\mathbf{5.41 \times 10^{-6}}$	> 0.05	-	$\mathbf{5.41 \times 10^{-6}}$	$\mathbf{5.41 \times 10^{-6}}$	> 0.05	-	> 0.05	> 0.05	> 0.05
MIXED	> 0.05	-	> 0.05	> 0.05	> 0.05	-	> 0.05	> 0.05	$\mathbf{8.83 \times 10^{-5}}$	-	> 0.05	$\mathbf{8.78 \times 10^{-5}}$
SG	> 0.05	$\mathbf{1.44 \times 10^{-2}}$	-	> 0.05	> 0.05	> 0.05	-	> 0.05	$\mathbf{8.88 \times 10^{-5}}$	$\mathbf{1.35 \times 10^{-4}}$	-	$\mathbf{8.83 \times 10^{-5}}$
ZHOU	$\mathbf{5.41 \times 10^{-6}}$	$\mathbf{5.41 \times 10^{-6}}$	$\mathbf{5.41 \times 10^{-6}}$	-	$\mathbf{5.41 \times 10^{-6}}$	$\mathbf{5.41 \times 10^{-6}}$	$\mathbf{5.41 \times 10^{-6}}$	-	> 0.05	> 0.05	> 0.05	-
	instance-100-2 / I_{HV}				**instance-100-2 / I_ϵ**				**instance-100-2 / I_O**			
	EX	MIXED	SG	ZHOU	EX	MIXED	SG	ZHOU	EX	MIXED	SG	ZHOU
EX	-	$\mathbf{5.41 \times 10^{-6}}$	$\mathbf{5.41 \times 10^{-6}}$	> 0.05	-	$\mathbf{5.41 \times 10^{-6}}$	$\mathbf{5.41 \times 10^{-6}}$	> 0.05	-	> 0.05	> 0.05	$\mathbf{9.62 \times 10^{-4}}$
MIXED	> 0.05	-	> 0.05	> 0.05	> 0.05	-	> 0.05	> 0.05	$\mathbf{8.93 \times 10^{-5}}$	-	> 0.05	$\mathbf{8.98 \times 10^{-5}}$
SG	> 0.05	$\mathbf{3.79 \times 10^{-5}}$	-	> 0.05	> 0.05	$\mathbf{5.25 \times 10^{-4}}$	-	> 0.05	$\mathbf{8.93 \times 10^{-5}}$	$\mathbf{8.83 \times 10^{-5}}$	-	$\mathbf{8.98 \times 10^{-5}}$
ZHOU	$\mathbf{5.41 \times 10^{-6}}$	$\mathbf{5.41 \times 10^{-6}}$	$\mathbf{5.41 \times 10^{-6}}$	-	$\mathbf{5.41 \times 10^{-6}}$	$\mathbf{5.41 \times 10^{-6}}$	$\mathbf{5.41 \times 10^{-6}}$	-	> 0.05	> 0.05	> 0.05	-
	instance-100-3 / I_{HV}				**instance-100-3 / I_ϵ**				**instance-100-3 / I_O**			
	EX	MIXED	SG	ZHOU	EX	MIXED	SG	ZHOU	EX	MIXED	SG	ZHOU
EX	-	$\mathbf{5.41 \times 10^{-6}}$	$\mathbf{5.41 \times 10^{-6}}$	> 0.05	-	$\mathbf{5.41 \times 10^{-6}}$	$\mathbf{5.41 \times 10^{-6}}$	> 0.05	-	> 0.05	> 0.05	$\mathbf{2.24 \times 10^{-2}}$
MIXED	> 0.05	-	> 0.05	> 0.05	> 0.05	-	> 0.05	> 0.05	$\mathbf{8.63 \times 10^{-5}}$	-	> 0.05	$\mathbf{8.63 \times 10^{-5}}$
SG	> 0.05	$\mathbf{1.44 \times 10^{-3}}$	-	> 0.05	> 0.05	$\mathbf{7.34 \times 10^{-3}}$	-	> 0.05	$\mathbf{8.88 \times 10^{-5}}$	$\mathbf{8.49 \times 10^{-5}}$	-	$\mathbf{8.88 \times 10^{-5}}$
ZHOU	$\mathbf{5.41 \times 10^{-6}}$	$\mathbf{5.41 \times 10^{-6}}$	$\mathbf{5.41 \times 10^{-6}}$	-	$\mathbf{5.41 \times 10^{-6}}$	$\mathbf{5.41 \times 10^{-6}}$	$\mathbf{5.41 \times 10^{-6}}$	-	> 0.05	> 0.05	> 0.05	-

Note: Bold font entries are significant to significance level $\alpha = 0.05$ (adjusted for multiple testing).

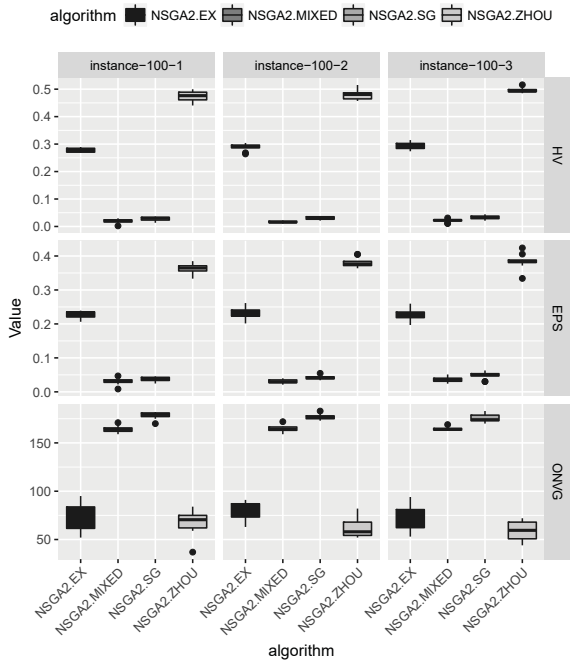

Figure 6: Boxplots of distributions of considered EMOA performance indicators: hypervolume I_{HV} (top row), ϵ-indicator I_ϵ (center row) and I_O (bottom row) for all considered instances.

to I_O. Since hypervolume- and ϵ-indicators are Pareto-compliant and ONVG is not, we conclude, that MIXED performs best in this scenario. To corroborate our assumption with statistical rigor, we perform pairwise statistical significance tests to the significance level $\alpha = 0.05$ for each pair of algorithms and each instance. We test the hypothesis pair

$$H_0^{A,B} : I^A(P) \geq I^B(P) \text{ vs. } H_1^{A,B} : I^A(P) < I^B(P)$$

for each instance P, indicator $I \in \{I_{\text{HV}}, I_\epsilon, I_O\}$ and algorithm pair A, B. The means of choice is the function `ecr::test`, which expects the above data frame as the single mandatory parameter[3]. Prior to testing, we apply some base R to shorten algorithm names for better readability in the resulting LaTeX-tables in Tab. 3.

```
unary = inds$unary
unary$algorithm = gsub(
  "NSGA2.", "",
  unary$algorithm, fixed = TRUE)
tests = ecr::test(
  unary,
  col.names = c("HV", "EPS", "ONVG"))
ecr::toLatex(tests)
```

The zero hypothesis for the interesting case of SG versus MIXED can be rejected for both I_{HV} and I_ϵ for all three problems, since the corresponding p-values are lower than α. I. e., there is enough statistcal evidence to accept the alternative hypothisis, which states that the medians of the distributions are lower for the MIXED operator.

4.4 Further notes

The tutorial covered the most important aspects and functionality and illustrated the toolchain by carrying out an example study on a bi-objective combinatorial problem. Of course, the majority of tools are not limited to the bi-objective case. In fact only scatterplots are possible up to three objectives. Note, that `ggplot2` unfortunatly does not support 3D scatterplots. Hence, 3D scatterplots rely on different packages which can be set with the `package` argument of the `ecr::plotScatter3d` function. For more details we refer the

[3]The function relies on reasonable and common defaults: non-parametric pairwise Wilcoxon rank sum test (no assumptions on the underlying distribution) with significance $\alpha = 0.05$. Moreover, multiple testing issues are ommited by p-value adjustments.

interested useR to the function documentations and the GitHub page https://github.com/jakobbossek/ecr2 of `ecr`.

5 CONCLUSION

This paper introduced functionality for performance assessment of multi-objective stochastic optimization algorithms shipped with the recent major update of the `ecr` package for the statistical programming language R. The methodology of this highly relevant topic was introduced with focus on practical application in a tutorial-like fashion utilizing a built-in example dataset. This way, following recommendations from the literature on multi-objective performance assessment, the most important tools and workflows were presented by example. There are many directions for future work: the agenda for the next release stipulates empirical attainment functions. Moreover, the set of built-in performance indicators is expandable and more alternatives and customization options for produced tablular output is desirable.

Acknowledgements

The authors acknowledge support from the *European Research Center for Information Systems* (ERCIS, https://www.ercis.org/).

REFERENCES

[1] Thomas Bartz-Beielstein, Christian Lasarczyk, and Mike Preuss. 2005. Sequential Parameter Optimization. In *Proceedings Congress on Evolutionary Computation 2005 (CEC'05)*. Edinburgh, Scotland, 1553. http://www.spotseven.de/wp-content/papercite-data/pdf/blp05.pdf

[2] Nicola Beume, Boris Naujoks, and Michael Emmerich. 2007. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research* 181, 3 (2007), 1653–1669. https://doi.org/10.1016/j.ejor.2006.08.008

[3] Mickael Binois and Victor Picheny. 2018. *GPareto: Gaussian Processes for Pareto Front Estimation and Optimization*. https://CRAN.R-project.org/package=GPareto R package version 1.1.1.

[4] Bernd Bischl, Jakob Richter, Jakob Bossek, Daniel Horn, Janek Thomas, and Michel Lang. 2017. *mlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions*. http://arxiv.org/abs/1703.03373

[5] S Bleuler, M Laumanns, L Thiele, and E Zitzler. 2003. PISAâĂŤa platform and programming language independent interface for search algorithms. In *Proceedings of International Conference on Evolutionary Multi-Criterion Optimization (EMO)*, C M Fonseca, P J Fleming, E Zitzler, K Deb, and L Thiele (Eds.). Springer, Berlin, Germany, 494–508. https://doi.org/10.1007/3-540-36970-8_35

[6] Jakob Bossek. 2017. ecr 2.0: A Modular Framework for Evolutionary Computation in R. In *Genetic and Evolutionary Computation Conference*. Berlin, Germany. https://doi.org/10.1145/3067695.3082470

[7] Jakob Bossek and Christian Grimme. 2017. A Pareto-Beneficial Sub-Tree Mutation for the Multi-Criteria Minimum Spanning Tree Problem. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, Honolulu, HI, USA, 3280–3287. https://doi.org/10.1109/SSCI.2017.8285183

[8] Jakob Bossek and Christian Grimme. 2017. An Extended Mutation-Based Priority-Rule Integration Concept for Multi-Objective Machine Scheduling. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, Honolulu, HI, USA, 3288–3295. https://doi.org/10.1109/SSCI.2017.8285224

[9] Jakob Bossek and Christian Grimme. 2018. Solving Scalarized Subproblems within Evolutionary Algorithms for Multi-Criteria Shortest Path Problems. In *Proceedings of the 12th International Conference on Learning and Intelligent Optimization (LION 2018)*. Springer International Publishing, Kalamata, Greece. accepted.

[10] Jakob Bossek and Heike Trautmann. 2016. Understanding characteristics of evolved instances for state-of-the-art inexact TSP solvers with maximum performance difference. In *AI*IA 2016 Advances in Artificial Intelligence*, G. Adorni, S. Cagnoni, M. Gori, and M. Maratea (Eds.), Vol. 10037 LNAI. Springer International Publishing, Genova, Italy, 3–12. https://doi.org/10.1007/978-3-319-49130-1_1

[11] Carlos A Coello Coello, Gary B Lamont, and David A Van Veldhuizen. 2006. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[12] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T Meyarivan. 2000. A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. In *Parallel Problem Solving from Nature PPSN VI*, Marc Schoenauer, Kalyanmoy Deb, Günther Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 849–858.

[13] Matt Dowle and Arun Srinivasan. 2017. *data.table: Extension of 'data.frame'*. https://CRAN.R-project.org/package=data.table R package version 1.10.4-3.

[14] N. Hansen and A. Jaszkiewicz. 2006. *Evaluating the quality of approximations to the non-dominated set.* Technical Report. Technical University of Denmark.

[15] Joshua Knowles and David Corne. 2002. On Metrics for Comparing Non-Dominated Sets. In *Proceedings of the 2002 Congress on Evolutionary Computation Conference (CEC02)*. Institute of Electrical and Electronics Engineers, Honolulu, HI, USA, 711–716.

[16] J Knowles, L Thiele, and E Zitzler. 2006. *A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers.* Technical Report. Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland. https://sop.tik.ee.ethz.ch/KTZ2005a.pdf

[17] Asep Maulana, Marios Kefalas, and Michael T. M. Emmerich. 2017. Immunization of Networks Using Genetic Algorithms and Multiobjective Metaheuristics. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, Honolulu, HI, USA, 2953–2960. https://doi.org/10.1109/SSCI.2017.8285368

[18] Olaf Mersmann. 2012. *emoa: Evolutionary Multiobjective Optimization Algorithms*. https://CRAN.R-project.org/package=emoa R package version 0.5-0.

[19] Olaf Mersmann. 2014. *mco: Multiple Criteria Optimization Algorithms and Related Functions*. https://CRAN.R-project.org/package=mco R package version 1.0-15.1.

[20] Adriano Passos. 2017. *moko: Multi-Objective Kriging Optimization*. https://CRAN.R-project.org/package=moko R package version 1.0.1.

[21] Serpil Sayın. 2000. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Mathematical Programming* 87, 3 (01 May 2000), 543–560. https://doi.org/10.1007/s101070050011

[22] V. T'kindt and J.-C. Billaut. 2006. *Multicriteria Scheduling: Theory, Models and Algorithms* (2nd ed.). Berlin Heidelberg.

[23] David Allen Van Veldhuizen. 1999. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Ph.D. Dissertation. Wright Patterson AFB, OH, USA. Advisor(s) Lamont, Gary B. AAI9928483.

[24] Hadley Wickham. 2009. *ggplot2: Elegant Graphics for Data Analysis*. Springer New York. http://ggplot2.org

[25] Hadley Wickham, Romain Francois, Lionel Henry, and Kirill MÃijller. 2017. *dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr R package version 0.7.4.

[26] E. Zitzler and L. Thiele. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* 3, 4 (1999), 257–271. https://doi.org/10.1109/4235.797969

[27] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca. 2003. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions of Evolutionary Computation* 7, 2 (2003), 117–132.