mathematics

MDPI

Article

Multi-Objective Optimization Benchmarking Using DSCTool

Peter Korošec * and Tome Eftimov

Computer Systems Department, Jožef Stefan Institute, Jamova cesta 39, SI-1000 Ljubljana, Slovenia; tome.eftimov@ijs.si

* Correspondence: peter.korosec@ijs.si

Received: 8 May 2020; Accepted: 21 May 2020; Published: 22 May 2020

Abstract: By performing data analysis, statistical approaches are highly welcome to explore the data. Nowadays with the increases in computational power and the availability of big data in different domains, it is not enough to perform exploratory data analysis (descriptive statistics) to obtain some prior insights from the data, but it is a requirement to apply higher-level statistics that also require much greater knowledge from the user to properly apply them. One research area where proper usage of statistics is important is multi-objective optimization, where the performance of a newly developed algorithm should be compared with the performances of state-of-the-art algorithms. In multi-objective optimization, we are dealing with two or more usually conflicting objectives, which result in high dimensional data that needs to be analyzed. In this paper, we present a web-service-based e-Learning tool called DSCTool that can be used for performing a proper statistical analysis for multi-objective optimization. The tool does not require any special statistics knowledge from the user. Its usage and the influence of a proper statistical analysis is shown using data taken from a benchmarking study performed at the 2018 IEEE CEC (The IEEE Congress on Evolutionary Computation) is appropriate. Competition on Evolutionary Many-Objective Optimization.

Keywords: multi-objective optimization; statistics; benchmarking; DSCTool

1. Introduction

Nowadays, comparing the performance of a newly developed multi-objective optimization algorithm involves calculating descriptive statistics such as means, medians, and standard deviations of the algorithm's performance. Recently, we have published several studies that show that calculating descriptive statistics is not enough to compare algorithms' performances, but require some high-level statistics [1–3].

In this paper, we will show how different benchmarking practices can potentially have big influences on the outcome of a statistical analysis performed on multi-objective optimization algorithms. Additionally, we will present a web-service-based e-Learning tool called DSCTool [4], which guides the user through all necessary steps needed to perform a proper statistical analysis. The DSCTool also reduces the requirement of additional statistical knowledge from the user, which includes knowing which conditions must be fulfilled to select a relevant and proper statistical test (e.g., parametric or nonparametric) [5]. The conditions that should be checked before selecting a relevant statistical test includes checking for data independence, normality, and homoscedasticity of variances.

In multi-objective optimization, there is no single solution that simultaneously optimizes each objective. Since the objectives are said to be conflicting, there exists a set of alternative solutions. Each solution that belongs to the set of alternative solutions is optimal in a manner that no other solution from the search space is superior to it when all objectives are considered. The question

that appears here is how to compare algorithms with regard to sets of solutions. To this end, many different performance metrics have been proposed, which map the solution sets to a set of real numbers. By using performance metrics, we can quantify the differences between solution sets. Further, this data can be used as input data that should be analyzed by applying some statistical tests.

To see the importance of making a proper statistical analysis in benchmarking studies that involve multi-objective optimizaton algorithms we performed analysis on the results presented at the 2018 IEEE CEC Competition on Evolutionary Many-Objective Optimization. To do this, the analyses are performed using the DSCTool, where each step from the DSCTool pipeline is explained in more detail. More details about the DSCTool are presented in [4].

The rest of the paper is organized as follows. In Section 2, the multi-objective optimization is shortly reintroduced, followed by Section 3 where two imporatant caveats of statistical analysis are presented. Section 4 reintroduces the DSCTool and Section 5 shows and compares different statistical analysis performed by using the DSCTool. Finally, the conclusions of the paper are presented in Section 6.

2. Multi-Objective Optimization

Multi-objective optimization is a research of multiple-criteria decision making, concerning with optimization problems that involve more than one objective, which should be optimized simultaneously. A multi-objective problem can be formulated as

$$min(f_1(x), f_2(x), ..., f_k(x)), x \in X$$

where $k \geq 2$ is the number of objectives and X is a set of feasible solutions. X must often satisfy some inequality constraints $g_i(x) \leq 0, i = 1, ..., l$ and equality constraints $h_j(x) = 0, i = 1, ..., m$. Since typical optimization algorithms are unconstrained search methods, dealing with constraints is a challenging task. Standard approaches [6] to handling constraints consists of penalty functions, objectivization of constraints violations, repair algorithms, separation of constraints and objectives, hybrid methods.

The objectives are usually conflicting, therefore there exists some kind of trade-off among objectives, i.e., we can only improve one objective at the expense of the other. As a consequence there is not one optimum solution for all objectives, but a set of solutions called Pareto optimal set, which are all considered optimal with respect to all objectives. There are many quality aspects with regard to which approximation sets are compared, such as closeness to the Pareto optimal set, and coverage of a wide range of diverse solutions. Quality is measured in terms of criteria that relates to properties of convergence and diversity. There have been many studies tackling this problem by using unary and binary quality indicators, which take one or two input sets that are mapped into a real number. However, one problem the arises is that there exists many quality indicators, for example, hypervolume (HV) [7], epsilon indicator (EI) [8], generational distance (GD) [9], inverse generational distance (IGD) [9] etc., that capture different information from the approximation set and its selection can greatly influence the outcome of the comparison study. For the interested reader Riquelme et al. [10] provide an in-depth review and analysis of 54 multi-objective-optimization metrics. One option to reduce the influence of the selection of a quality indicator, is to use ensemble learning, where the idea is to combine information gained from several quality indicators into one real value. With a suitable performance metric calculated, we then need to perform some statistical analysis to gain understanding of the compared algorithms performances.

3. Statistical Analysis

Frequentist statistical analysis can be done by exploratory data analysis, which involves calculating descriptive statistics (i.e., mean, median, standard deviation, etc.), or some higher-level statistics that are done by inferential statistics to explore relations between compared variables (e.g., hypothesis testing with statistical tests) [2]. No matter which statistical analysis is done, it is

very important to understand all its limitations and caveats because with its improper usage, wrong conclusions can be made. For example, improper usage comes from misunderstanding or not knowing the requirements when a certain statistical test can be applied. Recently, it was shown that outliers and small differences in the data can have negative impact on the results obtained from statistical tests [11].

3.1. Outliers

An outlier is an observation that lies outside of the distribution of the data [12]. If outliers are not properly handled, some deceptive statistics can be affected, which might not reflect the actual probability data distribution. This indicates that the statistical analysis shows that there is a statistical significance, while the probability distribution shows the opposite.

Outliers are data points that differ significantly from other data points in the data set and can cause serious problems in statistical analyses. For example, means are the most commonly used descriptive statistics that are involved in comparison studies because they are unbiased estimators. However, they are sensitive to outliers, so by using them for statistical analysis we inherently transfer this sensitivity to results of any analysis that originates from them. One option to reduce the influence of outliers is to use medians as a more robust statistic since they are less sensitive to outliers.

3.2. Small Differences in the Data

Though medians reduce the sensitivity to outliers, both means and medians can have negative impact on results obtained from applying statistical tests. Such impact can be observed when differences between means or medians are in some ϵ-neighbourhood. An ϵ-neighborhood is a range in function value space in which distance from a given number is less than some specified number ϵ. When the means or medians are not the same, but in some ϵ-neighborhood, they receive different rankings. However, if distributions of multiple runs are the same, indicating that there are no performance differences between the compared algorithms, this suggests that they should obtain the same rankings. On the contrary, it can also happen that the distributions of multiple runs are not the same, indicating that algorithms should obtain different rankings, but the means or medians are the same.

4. The DSCTool

The DSCTool is presented in [4] as an e-Learning tool with the goal to make all required scenarios for different performance evaluations of single- and multi-objective optimization algorithms. It guides the user through several steps of statistical analysis pipeline, starting from preparing and providing input data (optimisation algorithm results), then selecting the desired comparison scenario, and obtaining final result from it. In this way, the statistical knowledge required from the user is greatly reduced. Actually, the DSCTool allows users not to think about how the statistical analysis should be performed, but only to define the comparison scenario that is relevant for their experimental setup. Basically, the user must select a significance level used by statistical test and follow a pipeline for selecting the most commonly-used statistical tests in benchmarking evolutionary algorithms. For multi-objective optimization algorithm analysis, the user must provide data in form of one or more quality indicators, decide on significance level and which kind of ensemble (i.e., average, hierarchical majority vote, or data-driven) is desired for comparing data calculated by quality indicator. Following the pipeline is trivial as it will be shown in next sections.

The DSCTool is developed for the Deep Statistical Comparison (DSC) approach [2] and its variants that all have a benefit of providing robust statistical analysis with reduced influence of outliers and small differences in the data since a comparison is made on data distribution. In this paper, we will only show web services that implement DSC variants used with multi-objective optimization, i.e., the basic DSC ranking scheme used for comparing data for one quality indicator [13], and average, hierarchical majority vote, and data-driven ensembles that are used to fuse the information from more quality indicators [1,3].

The DSC ranking scheme is based on comparing distributions using some two-sample statistical test [14] with predefined significance level. The obtained ranking scheme is used to make a further statistical comparison. The DSC average ensemble is a ranking scheme that uses a set of user-selected quality indicators [1] to calculate a mean of DSC rankings for each quality indicator on specific problems [15]. The DSC hierarchical majority vote ensemble is a ranking scheme [1] that checks which algorithm wins in the most quality indicators or which algorithm is ranked the most number of times with the best DSC ranking on each benchmark problem separately. Finally, the DSC data-driven ensemble [3] is where the preference of each quality indicator is estimated using its entropy, which is calculated by the Shannon entropy weighted method [16]. The preference ranking organization method (PROMETHEE) [17] is then used to determine the rankings.

The DSCTool offers web services that are accessible from https://ws.ijs.si:8443/dsc-1.5/service/ and use the REST software architectural style. The web services are using JSON format to transfer input/output data. A more detailed information about using DSCTool can be accessed at https://ws.ijs.si:8443/dsc-1.5/documentation.pdf.

5. Experiments, Results, and Discussion

To show the usefulness and the power of DSCTool, the results presented at the 2018 IEEE CEC Competition on Evolutionary Many-Objective Optimization [18] are analyzed, where 10 algorithms were submitted for competition (i.e., AGE-II [19], AMPDEA, BCE-IBEA[20], CVEA3 [21], fastCAR [22], HHcMOEA [23], KnEA [24], RPEA [25], RSEA [26], and RVEA [27]). The competition provided 15 optimization problems (MaF01-MaF15) with 5, 10, and 15 objectives [28]. The obtained optimization results for each algorithm together with results can be accessed at https://github.com/ranchengcn/IEEE-CEC-MaOO-Competition/tree/master/2018. We would like to point here that we focus only on the issue of how to evaluate the performance (i.e., which statistical approach to be used), where the problem selection and experimental setup have been already solved and set by the competition organizers.

For performance metrics, the organizers have selected inverted generational distance (IGD), where 10,000 uniformly distributed reference points were sampled on the Pareto front and hypervolume (HV), where the population was normalized by the nadir point of Pareto front and Monte Carlo estimation method with 1,000,000 points was adopted. Each algorithm was executed 20 times on each problem with 5, 10, and 15 objectives, resulting in 900 approximation sets. Using these approximation sets, both quality indicators were calculated and the mean of each quality indicator was taken as a comparison metric for each problem and number of objectives. The algorithms were then ranked according to the comparison metric and the final score of the algorithm was determined as the sum of the reciprocal values of the rankings. In Tables 1 and 2, official competition ranking results for 5 and 10 objectives are presented for both quality indicators, respectively.

Using the tables, it can be seen that CVEA3 is the best performing algorithm due to obtained total ranking of 1 for both quality indicators in 5 and 10 objectives.

Further, statistical analysis on the same data is showed using the DSCTool. First, quality indicators should be calculated using the approximation sets obtained by the algorithms. In addition to IGD and HV, also generational distance (GD) and epsilon indicator (EI) were calculated. After all quality indicators are calculated, the data for each quality indicator must be organized in an appropriate JSON input for the rank web service that performs the DSC ranking scheme. The JSON inputs for each quality indicator and each number of objectives are available at http://cs.ijs.si/dl/dsctool/rank-json.zip. For our analysis, the two-sample Anderson–Darling test was selected to compare data distributions (since it is more powerful than Kolmogorov–Smirnov test). To obtain the DSC rankings for each quality indicator, the rank web service was executed with an appropriate JSON input. The result is a JSON response where rankings are calculated based on the DSC ranking scheme. The rankings results for all quality indicators for 5 and 10 objectives are provided in Tables 3–6, respectively.

Table 1. Official competition inverse generational distance (IGD) results.

(a) D = 5										
Problem	**AGE-II**	**AMPDEA**	**BCE-IBEA**	**CVEA3**	**fastCAR**	**HHcMOEA**	**KnEA**	**RPEA**	**RSEA**	**RVEA**
MaF01	1	4	3	5	6	8	2	9	7	10
MaF02	1	2	3	4	7	5	6	10	8	9
MaF03	4	3	8	1	2	6	9	10	7	5
MaF04	9	2	3	1	4	7	6	5	8	10
MaF05	4	9	1	2	5	8	3	10	7	6
MaF06	5	4	1	2	7	6	3	8	10	9
MaF07	6	4	2	1	7	5	3	10	8	9
MaF08	1	4	2	3	6	5	9	8	7	10
MaF09	1	5	6	2	3	4	10	8	9	7
MaF10	7	10	1	5	6	8	3	9	4	2
MaF11	8	6	5	4	9	7	3	10	2	1
MaF12	6	10	1	2	4	9	3	8	7	5
MaF13	2	9	4	1	5	3	6	8	7	10
MaF14	4	1	8	2	3	7	5	10	6	9
MaF15	2	3	8	1	4	5	10	9	7	6
Total	3	4	2	1	5	7	6	10	8	9
(b) D = 10										
Problem	**AGE-II**	**AMPDEA**	**BCE-IBEA**	**CVEA3**	**fastCAR**	**HHcMOEA**	**KnEA**	**RPEA**	**RSEA**	**RVEA**
MaF01	6	1	2	4	8	7	3	9	5	10
MaF02	7	5	3	2	8	4	1	6	10	9
MaF03	5	2	10	6	1	4	9	7	8	3
MaF04	9	1	6	2	4	8	5	3	7	10
MaF05	9	4	1	2	7	10	5	3	6	8
MaF06	5	2	9	1	4	7	10	3	8	6
MaF07	5	4	1	2	7	6	3	8	9	10
MaF08	2	6	1	3	9	4	5	8	7	10
MaF09	1	7	9	3	4	5	10	6	2	8
MaF10	8	9	1	6	2	7	3	10	5	4
MaF11	3	7	4	2	9	1	6	8	5	10
MaF12	10	9	2	1	6	7	5	4	8	3
MaF13	5	9	2	1	8	3	4	6	7	10
MaF14	6	1	9	2	4	8	10	5	7	3
MaF15	3	2	9	1	5	8	10	7	6	4
Total	4	2	3	1	5	6	7	8	9	10

Table 2. Official competition hypervolume (HV) results.

(a) D = 5										
Problem	**AGE-II**	**AMPDEA**	**BCE-IBEA**	**CVEA3**	**fastCAR**	**HHcMOEA**	**KnEA**	**RPEA**	**RSEA**	**RVEA**
MaF01	1	2	4	5	6	8	3	9	7	10
MaF02	7	4	3	1	8	10	2	6	5	9
MaF03	7	3	6	1	2	4	9	10	8	5
MaF04	6	5	8	1	7	9	2	3	4	10
MaF05	6	9	4	3	1	10	5	8	7	2
MaF06	2	4	1	3	6	7	5	9	8	10
MaF07	6	4	1	2	7	8	5	10	3	9
MaF08	1	4	3	2	6	7	8	9	5	10
MaF09	2	5	6	1	3	4	10	7	9	8
MaF10	8	10	3	1	9	7	4	5	2	6
MaF11	10	8	3	1	4	5	7	9	2	6
MaF12	7	10	5	1	3	9	2	8	6	4
MaF13	1	10	4	2	5	3	7	6	8	9
MaF14	4	2	8	1	3	6	5	9	7	10
MaF15	2	3	9	1	4	6	10	8	7	5
Total	3	5	2	1	4	8	9	10	7	9

Table 2. *Cont.*

(b) D = 10										
Problem	**AGE-II**	**AMPDEA**	**BCE-IBEA**	**CVEA3**	**fastCAR**	**HHcMOEA**	**KnEA**	**RPEA**	**RSEA**	**RVEA**
MaF01	6	4	2	7	3	9	5	8	1	10
MaF02	8	4	6	3	5	9	10	2	1	7
MaF03	6	1	8	2	4	3	10	7	9	5
MaF04	8	3	9	2	7	6	4	5	1	10
MaF05	9	4	2	3	1	10	5	7	8	6
MaF06	6	1	9	2	3	5	10	4	7	8
MaF07	10	3	7	5	4	9	8	2	1	6
MaF08	4	5	3	2	9	7	6	8	1	10
MaF09	1	6	9	3	4	5	10	7	2	8
MaF10	10	9	6	1	3	7	5	4	2	8
MaF11	10	1	4	2	6	5	7	9	3	8
MaF12	10	9	6	1	3	8	2	7	4	5
MaF13	7	10	4	1	3	5	6	2	9	8
MaF14	6	2	8	1	4	7	10	5	9	3
MaF15	5	1	7	2	4	6	9	10	8	3
Total	2	6	1	3	7	6	10	5	4	8

Table 3. DSCTool rankings for IGD.

(a) D = 5										
Problem	**AGE-II**	**AMPDEA**	**BCEIBEA**	**CVEA3**	**fastCAR**	**HHcMOEA**	**KnEA**	**RPEA**	**RSEA**	**RVEA**
MaF01	2.0	5.5	3.0	4.0	1.0	7.5	5.5	9.0	7.5	10.0
MaF02	1.5	5.0	1.5	3.0	7.0	7.0	4.0	10.0	7.0	9.0
MaF03	10.0	9.0	3.0	4.0	8.0	1.0	7.0	6.0	5.0	2.0
MaF04	9.0	2.0	7.0	1.0	3.0	8.0	5.0	4.0	6.0	10.0
MaF05	4.0	8.0	5.0	3.0	2.0	9.0	6.0	10.0	7.0	1.0
MaF06	3.0	4.0	1.0	2.0	7.0	6.0	5.0	8.0	10.0	9.0
MaF07	4.0	5.0	1.0	2.0	7.0	6.0	3.0	10.0	8.0	9.0
MaF08	2.0	4.0	1.0	3.0	5.0	6.0	9.0	8.0	7.0	10.0
MaF09	5.0	1.0	6.0	3.0	2.0	4.0	10.0	8.0	9.0	7.0
MaF10	8.0	10.0	1.0	3.0	6.0	7.0	2.0	9.0	4.0	5.0
MaF11	9.0	6.0	6.0	3.0	1.0	8.0	3.0	10.0	6.0	3.0
MaF12	5.5	10.0	5.5	3.5	1.0	8.0	3.5	8.0	8.0	2.0
MaF13	9.0	10.0	1.0	8.0	5.0	3.0	2.0	7.0	4.0	6.0
MaF14	4.0	1.0	8.0	2.0	3.0	7.0	5.0	10.0	6.0	9.0
MaF15	3.0	4.0	2.0	6.0	5.0	1.0	10.0	9.0	8.0	7.0
Total	4	6	2	1	3	7	5	10	9	8
(b) D = 10										
Problem	**AGE-II**	**AMPDEA**	**BCEIBEA**	**CVEA3**	**fastCAR**	**HHcMOEA**	**KnEA**	**RPEA**	**RSEA**	**RVEA**
MaF01	4.0	1.0	5.0	2.0	9.0	7.0	3.0	8.0	6.0	10.0
MaF02	2.0	4.0	1.0	3.0	7.0	5.0	9.0	6.0	8.0	10.0
MaF03	10.0	7.0	2.0	8.0	9.0	3.0	1.0	6.0	4.0	5.0
MaF04	9.0	1.0	7.0	2.0	3.0	8.0	5.0	4.0	6.0	10.0
MaF05	9.0	4.0	1.0	2.0	6.0	10.0	5.0	3.0	8.0	7.0
MaF06	5.0	10.0	4.0	9.0	7.0	2.0	3.0	8.0	1.0	6.0
MaF07	6.0	2.0	4.0	1.0	7.0	5.0	3.0	9.0	10.0	8.0
MaF08	3.0	6.0	1.0	2.0	9.0	4.0	5.0	8.0	7.0	10.0
MaF09	10.0	2.0	1.0	7.0	9.0	4.0	3.0	5.0	6.0	8.0
MaF10	10.0	9.0	1.0	5.0	3.0	7.0	2.0	8.0	4.0	6.0
MaF11	10.0	3.0	1.0	4.0	8.0	7.0	2.0	9.0	5.0	6.0
MaF12	10.0	9.0	4.0	3.0	1.0	8.0	5.0	6.0	7.0	2.0
MaF13	4.0	10.0	1.0	5.0	6.0	3.0	7.0	8.0	2.0	9.0
MaF14	6.0	2.0	9.0	1.0	3.0	8.0	10.0	5.0	7.0	4.0
MaF15	5.0	4.0	9.0	7.0	6.0	1.0	10.0	3.0	2.0	8.0
Total	9	4	1	2	7	5	3	8	6	10

Table 4. DSCTool rankings for HV.

(a) D = 5										
Problem	**AGE-II**	**AMPDEA**	**BCEIBEA**	**CVEA3**	**fastCAR**	**HHcMOEA**	**KnEA**	**RPEA**	**RSEA**	**RVEA**
MaF01	4.5	3.0	4.5	1.0	2.0	7.5	7.5	7.5	7.5	10.0
MaF02	9.0	4.0	3.0	2.0	7.0	10.0	1.0	6.0	5.0	8.0
MaF03	7.0	3.0	10.0	3.0	6.0	9.0	8.0	3.0	3.0	3.0
MaF04	6.5	1.5	4.5	1.5	6.5	4.5	9.0	3.0	10.0	8.0
MaF05	8.0	10.0	6.0	3.0	2.0	7.0	4.0	9.0	5.0	1.0
MaF06	7.0	3.0	1.0	2.0	6.0	5.0	4.0	8.0	10.0	9.0
MaF07	2.0	5.0	5.0	1.0	8.0	3.0	5.0	10.0	9.0	7.0
MaF08	4.0	3.0	2.0	1.0	6.0	6.0	9.0	6.0	8.0	10.0
MaF09	7.0	6.0	3.0	1.0	5.0	4.0	10.0	2.0	8.0	9.0
MaF10	8.0	10.0	3.0	1.0	9.0	6.0	5.0	4.0	2.0	7.0
MaF11	9.0	10.0	4.0	1.0	5.0	2.0	7.0	8.0	3.0	6.0
MaF12	7.0	10.0	6.0	1.0	4.0	8.0	3.0	9.0	2.0	5.0
MaF13	4.0	10.0	7.0	1.0	5.0	3.0	6.0	2.0	9.0	8.0
MaF14	9.0	8.0	4.0	2.0	7.0	3.0	5.0	6.0	1.0	10.0
MaF15	5.0	2.0	3.0	1.0	7.0	4.0	6.0	8.0	10.0	9.0
Total	9	5	2	1	4	3	6	7	8	10
(b) D = 10										
Problem	**AGE-II**	**AMPDEA**	**BCEIBEA**	**CVEA3**	**fastCAR**	**HHcMOEA**	**KnEA**	**RPEA**	**RSEA**	**RVEA**
MaF01	4.5	2.0	6.5	1.0	9.0	8.0	3.0	4.5	6.5	10.0
MaF02	9.0	4.0	5.0	2.0	6.0	7.0	10.0	3.0	1.0	8.0
MaF03	9.0	1.0	7.0	4.0	2.0	5.0	6.0	10.0	8.0	3.0
MaF04	7.0	1.0	4.0	2.0	6.0	5.0	9.0	3.0	10.0	8.0
MaF05	10.0	6.0	4.0	2.0	3.0	9.0	1.0	7.0	5.0	8.0
MaF06	7.0	1.0	9.0	4.0	3.0	8.0	10.0	2.0	5.0	6.0
MaF07	6.0	5.0	7.0	4.0	8.0	1.0	2.0	10.0	9.0	3.0
MaF08	6.0	5.0	4.0	2.0	9.0	3.0	7.0	1.0	8.0	10.0
MaF09	7.0	6.0	8.5	1.0	3.5	3.5	10.0	2.0	5.0	8.5
MaF10	10.0	9.0	8.0	1.0	5.0	3.0	6.0	4.0	2.0	7.0
MaF11	10.0	1.0	4.0	2.0	6.0	5.0	8.0	9.0	3.0	7.0
MaF12	10.0	8.5	6.0	1.5	4.5	8.5	3.0	7.0	1.5	4.5
MaF13	7.0	9.5	5.0	2.0	3.5	3.5	7.0	1.0	7.0	9.5
MaF14	9.0	10.0	1.0	4.0	7.0	5.0	2.0	6.0	8.0	3.0
MaF15	6.0	2.0	4.0	1.0	7.0	5.0	10.0	3.0	9.0	8.0
Total	10	2	6	1	5	4	8	3	7	9

Table 5. DSCTool rankings for EI.

(a) D = 5										
Problem	**AGE-II**	**AMPDEA**	**BCEIBEA**	**CVEA3**	**fastCAR**	**HHcMOEA**	**KnEA**	**RPEA**	**RSEA**	**RVEA**
MaF01	1.0	2.0	4.0	3.0	5.0	6.0	8.0	9.0	7.0	10.0
MaF02	1.0	4.5	2.5	2.5	7.5	7.5	4.5	7.5	7.5	10.0
MaF03	1.0	3.0	8.0	5.0	2.0	6.0	7.0	10.0	9.0	4.0
MaF04	1.0	4.0	7.0	3.0	2.0	6.0	9.0	5.0	10.0	8.0
MaF05	1.0	9.0	3.0	2.0	5.0	8.0	6.0	10.0	7.0	4.0
MaF06	1.0	4.0	2.0	3.0	7.0	6.0	5.0	8.0	10.0	9.0
MaF07	1.0	6.0	4.0	3.0	8.0	5.0	2.0	10.0	9.0	7.0
MaF08	1.0	4.0	2.0	3.0	5.5	5.5	9.5	7.5	7.5	9.5
MaF09	1.0	5.0	7.0	2.0	4.0	3.0	10.0	8.0	9.0	6.0
MaF10	2.0	10.0	4.0	5.0	6.0	9.0	3.0	7.0	1.0	8.0
MaF11	1.0	10.0	7.0	6.0	4.0	8.0	3.0	9.0	2.0	5.0
MaF12	1.0	10.0	5.0	3.0	2.0	9.0	6.0	8.0	7.0	4.0
MaF13	1.0	10.0	4.0	2.0	5.0	3.0	6.0	8.0	7.0	9.0
MaF14	2.0	1.0	8.0	3.0	4.0	6.0	5.0	9.0	7.0	10.0
MaF15	2.0	4.0	8.0	1.0	3.0	6.0	10.0	9.0	7.0	5.0
Total	1	5	4	2	3	6	6	10	8	9

Table 5. *Cont.*

(b) D = 10										
Problem	**AGE-II**	**AMPDEA**	**BCEIBEA**	**CVEA3**	**fastCAR**	**HHcMOEA**	**KnEA**	**RPEA**	**RSEA**	**RVEA**
MaF01	3.0	1.0	7.0	2.0	9.0	4.0	6.0	5.0	8.0	10.0
MaF02	1.0	3.0	3.0	7.0	7.0	7.0	10.0	3.0	7.0	7.0
MaF03	1.0	3.0	10.0	6.0	5.0	3.0	9.0	7.0	8.0	3.0
MaF04	1.0	3.0	8.0	4.0	2.0	5.0	9.0	7.0	10.0	6.0
MaF05	4.0	7.0	9.0	8.0	1.0	10.0	3.0	6.0	5.0	2.0
MaF06	5.0	3.0	9.0	1.0	4.0	8.0	10.0	2.0	7.0	6.0
MaF07	1.0	6.0	7.0	5.0	8.0	3.0	4.0	10.0	9.0	2.0
MaF08	2.0	5.0	1.0	3.0	9.0	4.0	8.0	6.0	7.0	10.0
MaF09	4.0	7.0	9.0	1.0	2.0	5.0	10.0	6.0	3.0	8.0
MaF10	2.0	9.5	5.5	9.5	5.5	5.5	5.5	8.0	2.0	2.0
MaF11	1.0	6.0	7.0	10.0	3.0	9.0	4.0	8.0	5.0	2.0
MaF12	1.0	10.0	7.0	5.0	4.0	9.0	2.0	6.0	8.0	3.0
MaF13	4.0	9.0	2.0	1.0	8.0	3.0	5.0	6.0	7.0	10.0
MaF14	6.0	1.0	9.0	2.0	4.0	8.0	10.0	5.0	7.0	3.0
MaF15	3.0	1.0	9.0	2.0	4.0	8.0	10.0	6.0	7.0	5.0
Total	1	3	9	2	4	7	10	6	8	5

Table 6. DSCTool rankings for GD.

(a) D = 5										
Problem	**AGE-II**	**AMPDEA**	**BCEIBEA**	**CVEA3**	**fastCAR**	**HHcMOEA**	**KnEA**	**RPEA**	**RSEA**	**RVEA**
MaF01	2.0	1.0	7.0	3.0	6.0	9.0	4.0	8.0	5.0	10.0
MaF02	1.5	1.5	4.5	6.5	8.5	10.0	3.0	6.5	4.5	8.5
MaF03	5.0	2.0	8.0	4.0	1.0	10.0	7.0	6.0	3.0	9.0
MaF04	1.0	6.0	8.0	2.0	4.0	9.0	5.0	7.0	3.0	10.0
MaF05	4.0	1.0	6.0	9.0	2.0	10.0	7.0	5.0	8.0	3.0
MaF06	1.0	2.5	4.5	2.5	4.5	7.5	7.5	9.0	6.0	10.0
MaF07	1.0	9.0	4.0	5.0	7.0	8.0	6.0	2.0	3.0	10.0
MaF08	1.0	9.0	8.0	2.0	6.0	5.0	3.0	4.0	7.0	10.0
MaF09	1.0	10.0	7.0	2.0	5.0	4.0	8.0	6.0	3.0	9.0
MaF10	1.0	10.0	6.0	5.0	9.0	7.0	4.0	2.0	3.0	8.0
MaF11	1.0	8.0	10.0	6.0	3.0	8.0	8.0	2.0	5.0	4.0
MaF12	2.0	10.0	4.0	1.0	8.0	9.0	3.0	7.0	5.0	6.0
MaF13	1.0	3.0	9.0	2.0	6.0	10.0	8.0	4.0	7.0	5.0
MaF14	9.0	2.0	7.0	1.0	5.0	10.0	6.0	4.0	3.0	8.0
MaF15	3.5	7.0	9.0	1.5	3.5	7.0	10.0	7.0	1.5	5.0
Total	1	6	8	2	4	10	7	5	3	9
(b) D = 10										
Problem	**AGE-II**	**AMPDEA**	**BCEIBEA**	**CVEA3**	**fastCAR**	**HHcMOEA**	**KnEA**	**RPEA**	**RSEA**	**RVEA**
MaF01	6.0	1.5	8.0	5.0	1.5	10.0	4.0	7.0	3.0	9.0
MaF02	2.0	2.0	5.0	9.0	8.0	10.0	6.0	4.0	7.0	2.0
MaF03	1.0	4.0	10.0	3.0	2.0	8.0	9.0	5.0	7.0	6.0
MaF04	4.0	6.0	8.0	3.0	2.0	10.0	5.0	7.0	1.0	9.0
MaF05	3.0	1.0	4.0	9.0	2.0	10.0	7.0	6.0	8.0	5.0
MaF06	4.0	1.0	7.0	4.0	4.0	7.0	9.0	2.0	7.0	10.0
MaF07	1.0	7.5	3.5	3.5	7.5	10.0	7.5	3.5	3.5	7.5
MaF08	1.0	9.0	8.0	3.0	2.0	6.0	7.0	5.0	4.0	10.0
MaF09	1.0	8.0	9.0	4.0	2.0	6.0	7.0	5.0	3.0	10.0
MaF10	9.0	10.0	8.0	6.0	5.0	4.0	7.0	1.0	3.0	2.0
MaF11	1.0	6.0	9.0	7.0	4.0	10.0	8.0	3.0	5.0	2.0
MaF12	4.0	3.0	5.0	7.0	9.0	10.0	6.0	2.0	1.0	8.0
MaF13	10.0	2.0	8.0	5.0	6.0	7.0	4.0	1.0	9.0	3.0
MaF14	7.0	3.0	9.0	1.0	2.0	10.0	8.0	6.0	4.0	5.0
MaF15	4.0	6.0	9.0	3.0	2.0	10.0	8.0	7.0	5.0	1.0
Total	1	4	9	6	2	10	8	3	5	7

First, let us compare results obtained in the official competition (see Tables 1 and 2) and the results obtained by DSCTool, where comparisons are done using only one quality indicator (see Tables 3–6).

Here we would like to remind the reader that competition ranking is based on simple statistics in the form of the mean value obtained for a specific quality indicator, while DSCTool uses the DSC approach in which the comparison is based on the distribution of quality indicator values from several runs. Quickly one can see that for some algorithms the obtained ranking is the same (e.g., for algorithm CVEA3 on 5 objectives both approaches returned ranking 1 according to IGD and HV quality indicator), while for some others there is a large difference between the rankings (e.g., for algorithm AGE-II on 5 objectives and using HV quality indicator competition ranked it as 3rd, while DSCTool ranked it as 9th). Since the paper is not about comparing different statistical approaches, we will not go into details. Nevertheless, we would like to remind the reader that though mean values are unbiased estimators, they can be heavily influenced by outliers and some small differences in the data. For further details on this topic we refer the reader to [2]. Therefore, by only using different (in our case more powerful) statistics, conclusions obtained from the rankings can be drastically changed. In our experiment, two additional quality indicators (i.e., GD and EI) were calculated on purpose, so the influence of its selection can be even better seen. Looking at Tables 3–6, drastic changes in rankings can be observed. Again looking at algorithm AGE-II it can be seen that if only EI and GD would be used, the AGE-II would change from the average performing algorithm to the best one, also outperforming algorithm CVEA3 in all cases.

Though it is well known that the selection of the quality indicator can have a big influence on the statistical analysis outcome, this influence can be also clearly observed in Tables 3–6. In such situations, it is better to use an ensemble of several quality indicators and estimate performance according to all of them. The DSCTool provides the ensemble service, which calculates rankings according to inputs from several quality indicators. To do this, ranking results from running the rank web service on all quality indicators should be taken and used as inputs to the ensemble service. In addition to all quality indicator data that needs to be put into a proper JSON form, one needs to decide on which ensemble technique to be used. The DSCTool provides three ensemble options, namely: average, hierarchical, and data-driven. When using the average method, rankings are simply averaged, with the hierarchical method, the input rankings are looked at from a hierarchical viewpoint where the algorithm with the highest rankings obtains the best ensemble ranking, and the data-driven method, where the information gain provided by the quality indicators is also taken into account to determine the obtained ensemble ranking. The examples of the JSON input for the average ensemble method can be found at http://cs.ijs.si/dl/dsctool/M05-average.json and http://cs.ijs.si/dl/dsctool/M10-average.json for 5 and 10 objectives, respectively. To prepare the JSON input for hierarchical or data-driven ensemble, the name of the method should be modified/changed from average to hierarchical or data-driven, while everything else remains the same. Further, to obtain the ensemble rankings, the ensemble service was executed with an appropriate JSON input. The rankings obtained from ensemble services in 5 and 10 objectives are shown in Tables 7–9. Unsurprisingly, it can be again seen that the rankings have changed with respect to individual rankings shown in Tables 3–6. Now, the rankings no longer represent individual quality indicators, but a fusion of the information by all of them. All ensembles provide similar ranking results, but there are still some differences between them. So, the selection of an ensemble method can also have an influence and must be made with great care. If we are interested purely in average performance, average should be selected, if we are interested in most often high-performing algorithm, hierarchical should be selected (this is also recommended for dynamic optimization), while if we care about which algorithm seems the most relevant in general, the data-driven ensemble method should be selected.

Table 7. DSCTool average ensemble rankings.

(a) D = 5										
Problem	**AGE-II**	**AMPDEA**	**BCEIBEA**	**CVEA3**	**fastCAR**	**HHcMOEA**	**KnEA**	**RPEA**	**RSEA**	**RVEA**
MaF01	2.25	2.75	4.50	2.75	3.50	7.00	5.75	8.00	6.25	10.00
MaF02	3.00	3.50	2.50	3.25	6.75	8.00	3.00	7.00	5.25	8.75
MaF03	5.75	3.75	7.25	3.50	4.25	6.50	7.25	5.75	4.50	4.00
MaF04	4.25	3.25	6.50	1.75	3.75	6.75	7.00	4.75	7.25	9.00
MaF05	4.25	7.00	5.00	4.25	2.75	8.50	5.75	8.50	6.75	2.25
MaF06	3.00	3.25	2.00	2.25	6.00	6.00	5.25	8.25	9.00	9.25
MaF07	2.00	6.00	3.25	2.75	7.50	5.50	3.75	8.00	7.25	8.25
MaF08	2.00	5.00	3.25	2.25	5.25	5.25	7.50	6.00	7.25	9.75
MaF09	3.50	5.50	5.75	2.00	4.00	3.75	9.50	6.00	7.25	7.75
MaF10	4.75	10.00	3.50	3.50	7.50	7.25	3.50	5.50	2.50	7.00
MaF11	5.00	8.00	6.50	3.75	3.25	6.25	4.75	7.25	3.75	4.25
MaF12	3.75	10.00	5.00	2.00	3.75	8.25	3.75	7.75	5.25	4.25
MaF13	3.75	8.25	5.25	3.25	5.25	4.75	5.50	5.25	6.75	7.00
MaF14	6.00	3.00	6.75	2.00	4.75	6.50	5.25	7.25	4.25	9.25
MaF15	3.25	4.00	5.50	2.25	4.50	4.25	9.00	8.00	6.50	6.50
Total	2	5	3	1	4	8	6	9	7	10
(b) D = 10										
Problem	**AGE-II**	**AMPDEA**	**BCEIBEA**	**CVEA3**	**fastCAR**	**HHcMOEA**	**KnEA**	**RPEA**	**RSEA**	**RVEA**
MaF01	4.25	1.25	6.50	2.50	7.00	7.25	4.00	6.00	5.75	9.75
MaF02	3.25	2.75	3.25	4.75	6.50	6.75	8.75	3.75	5.25	6.00
MaF03	5.25	3.50	7.25	5.25	4.50	4.50	6.25	7.00	6.75	4.00
MaF04	5.25	2.75	6.75	2.75	3.25	7.00	7.00	5.25	6.75	8.25
MaF05	6.50	4.50	4.50	5.25	3.00	9.75	4.00	5.50	6.50	5.50
MaF06	5.00	3.75	7.00	4.25	4.25	6.00	8.00	3.50	4.75	7.00
MaF07	3.50	4.75	5.00	3.00	7.25	4.75	3.75	7.75	7.50	4.75
MaF08	3.00	6.25	3.50	2.50	7.25	4.25	6.75	5.00	6.50	10.00
MaF09	5.50	5.75	6.75	3.25	4.00	4.50	7.50	4.50	4.25	8.50
MaF10	7.50	9.25	5.25	5.25	4.25	4.50	4.75	5.25	2.50	4.00
MaF11	5.50	4.00	5.25	5.75	5.25	7.75	5.50	7.25	4.50	4.25
MaF12	6.25	7.50	5.50	4.00	4.50	8.75	4.00	5.25	4.25	4.25
MaF13	6.00	7.50	4.00	3.25	5.75	4.00	5.50	4.00	6.00	7.75
MaF14	7.00	4.00	7.00	2.00	4.00	7.75	7.50	5.50	6.50	3.75
MaF15	4.50	3.25	7.75	3.25	4.75	6.00	9.50	4.75	5.75	5.50
Total	4	2	7	1	3	10	8	5	6	9

Table 8. DSCTool hierarchical ensemble rankings.

(a) D = 5										
Problem	**AGE-II**	**AMPDEA**	**BCEIBEA**	**CVEA3**	**fastCAR**	**HHcMOEA**	**KnEA**	**RPEA**	**RSEA**	**RVEA**
MaF01	1.0	2.0	5.0	4.0	3.0	8.0	6.0	9.0	7.0	10.0
MaF02	1.0	4.0	2.0	5.0	8.0	9.0	3.0	7.0	6.0	10.0
MaF03	6.0	1.0	9.0	5.0	3.0	8.0	10.0	7.0	4.0	2.0
MaF04	2.0	3.0	8.0	1.0	4.0	7.0	9.0	5.0	6.0	10.0
MaF05	2.0	3.0	6.0	5.0	4.0	10.0	7.0	9.0	8.0	1.0
MaF06	2.0	4.0	1.0	3.0	6.0	7.0	5.0	9.0	8.0	10.0
MaF07	1.0	8.0	3.0	2.0	9.0	6.0	4.0	5.0	7.0	10.0
MaF08	1.0	4.0	2.0	3.0	7.5	7.5	5.0	6.0	9.0	10.0
MaF09	1.0	3.0	7.0	2.0	4.0	6.0	10.0	5.0	8.0	9.0
MaF10	2.0	10.0	3.0	4.0	8.0	9.0	5.0	6.0	1.0	7.0
MaF11	1.0	10.0	9.0	2.0	3.0	7.0	5.0	8.0	4.0	6.0
MaF12	3.0	10.0	7.0	1.0	2.0	9.0	6.0	8.0	5.0	4.0
MaF13	1.0	7.0	3.0	2.0	9.0	6.0	5.0	4.0	8.0	10.0
MaF14	4.0	1.0	8.0	2.0	5.0	6.0	9.0	7.0	3.0	10.0
MaF15	4.0	6.0	5.0	1.0	7.0	2.0	10.0	9.0	3.0	8.0
Total	1	3	4	2	5	9	7	8	6	10

Table 8. *Cont.*

(b) D = 10										
Problem	**AGE-II**	**AMPDEA**	**BCEIBEA**	**CVEA3**	**fastCAR**	**HHcMOEA**	**KnEA**	**RPEA**	**RSEA**	**RVEA**
MaF01	5.0	1.0	9.0	2.0	3.0	8.0	4.0	7.0	6.0	10.0
MaF02	1.0	2.0	3.0	7.0	9.0	8.0	10.0	6.0	4.0	5.0
MaF03	1.0	2.0	7.0	8.0	4.0	6.0	3.0	10.0	9.0	5.0
MaF04	2.0	1.0	7.0	4.0	5.0	8.0	9.0	6.0	3.0	10.0
MaF05	7.0	4.0	3.0	5.0	1.0	10.0	2.0	8.0	9.0	6.0
MaF06	7.0	1.0	9.0	2.0	6.0	5.0	8.0	4.0	3.0	10.0
MaF07	1.0	7.0	6.0	2.0	10.0	3.0	4.0	9.0	8.0	5.0
MaF08	2.0	8.0	1.0	4.0	5.0	6.0	9.0	3.0	7.0	10.0
MaF09	2.0	6.0	3.0	1.0	4.0	8.0	9.0	5.0	7.0	10.0
MaF10	6.0	10.0	3.5	5.0	9.0	8.0	7.0	3.5	1.0	2.0
MaF11	1.0	2.0	3.0	5.0	7.0	10.0	6.0	9.0	8.0	4.0
MaF12	4.0	8.0	9.0	2.0	3.0	10.0	6.0	7.0	1.0	5.0
MaF13	9.0	5.0	3.0	2.0	7.0	6.0	10.0	1.0	4.0	8.0
MaF14	10.0	2.0	3.0	1.0	4.0	9.0	5.0	8.0	7.0	6.0
MaF15	8.0	2.0	9.0	1.0	5.0	4.0	10.0	7.0	6.0	3.0
Total	3	2	4	1	5	10	9	7	6	8

Table 9. DSCTool data-driven ensemble rankings.

(a) D = 5										
Problem	**AGE-II**	**AMPDEA**	**BCEIBEA**	**CVEA3**	**fastCAR**	**HHcMOEA**	**KnEA**	**RPEA**	**RSEA**	**RVEA**
MaF01	1.0	3.0	5.0	2.0	4.0	8.0	6.0	9.0	7.0	10.0
MaF02	3.0	5.0	1.0	4.0	8.0	9.0	2.0	7.0	6.0	10.0
MaF03	6.0	2.0	10.0	1.0	3.0	8.0	9.0	7.0	5.0	4.0
MaF04	4.0	2.0	6.0	1.0	3.0	7.0	8.0	5.0	9.0	10.0
MaF05	4.0	8.0	5.0	3.0	2.0	9.5	6.0	9.5	7.0	1.0
MaF06	3.0	4.0	1.0	2.0	7.0	6.0	5.0	8.0	9.0	10.0
MaF07	1.0	6.0	3.0	2.0	8.0	5.0	4.0	9.0	7.0	10.0
MaF08	1.0	4.0	3.0	2.0	5.5	5.5	9.0	7.0	8.0	10.0
MaF09	2.0	5.0	6.0	1.0	4.0	3.0	10.0	7.0	8.0	9.0
MaF10	5.0	10.0	4.0	2.5	9.0	8.0	2.5	6.0	1.0	7.0
MaF11	5.0	10.0	8.0	3.0	1.0	7.0	6.0	9.0	2.0	4.0
MaF12	3.0	10.0	6.0	1.0	2.0	9.0	4.0	8.0	7.0	5.0
MaF13	2.0	10.0	4.5	1.0	4.5	3.0	7.0	6.0	8.0	9.0
MaF14	6.0	2.0	8.0	1.0	4.0	7.0	5.0	9.0	3.0	10.0
MaF15	2.0	3.0	6.0	1.0	5.0	4.0	10.0	9.0	8.0	7.0
Total	2	5	4	1	3	8	6	9	7	10
(b) D = 10										
Problem	**AGE-II**	**AMPDEA**	**BCEIBEA**	**CVEA3**	**fastCAR**	**HHcMOEA**	**KnEA**	**RPEA**	**RSEA**	**RVEA**
MaF01	4.0	1.0	7.0	2.0	8.0	9.0	3.0	6.0	5.0	10.0
MaF02	3.0	1.0	2.0	5.0	8.0	9.0	10.0	4.0	6.0	7.0
MaF03	5.0	1.0	10.0	6.0	3.0	4.0	7.0	9.0	8.0	2.0
MaF04	4.5	1.5	6.0	1.5	3.0	8.0	9.0	4.5	7.0	10.0
MaF05	8.5	3.0	4.0	5.0	1.0	10.0	2.0	7.0	8.5	6.0
MaF06	6.0	2.0	9.0	3.0	4.0	7.0	10.0	1.0	5.0	8.0
MaF07	2.0	5.5	7.0	1.0	8.0	4.0	3.0	10.0	9.0	5.5
MaF08	2.0	6.0	3.0	1.0	9.0	4.0	8.0	5.0	7.0	10.0
MaF09	6.0	7.0	8.0	1.0	2.0	5.0	9.0	4.0	3.0	10.0
MaF10	9.0	10.0	8.0	7.0	3.0	4.0	5.0	6.0	1.0	2.0
MaF11	6.5	1.0	4.0	8.0	5.0	10.0	6.5	9.0	3.0	2.0
MaF12	8.0	9.0	7.0	2.0	5.0	10.0	1.0	6.0	4.0	3.0
MaF13	7.5	9.0	3.0	1.0	6.0	4.0	5.0	2.0	7.5	10.0
MaF14	7.0	3.5	8.0	1.0	3.5	10.0	9.0	5.0	6.0	2.0
MaF15	3.0	1.0	9.0	2.0	4.5	8.0	10.0	4.5	7.0	6.0
Total	4	2	8	1	3	10	9	5	6	7

The algorithms rankings obtained on each benchmark problem consist of statistical significance that can be used to compare the algorithms only on that specific problem (or performing single

problem analysis). However, if we are interested in a more general conclusion, or to compare the algorithms using the set of all benchmark problems (or multiple-problem analysis), it is not enough only to look at the mean ranking of the algorithms obtained from all benchmark problems. In such situations, we should analyze the data by applying some paired statistical test. For this purpose an omnibus test must be performed, which is implemented in DSCTool as an omnibus web service. The omnibus test can be performed on the results from any of the above tables. When using the DSCTool rank or ensemble web service we receive a JSON result that consists of the rankings for all algorithms and all benchmark problems (which were shown in Tables 3–9). Additionally, it provides us the information if the parametric tests can be applied on our data and which statistical tests are relevant for our data. In our cases, the Friedman test was recommended as an appropriate one. To make a general conclusion, we continue by using it with a significance level of 0.05. Instead of showing the results of all obtained rankings (presented in the tables), only one scenario was selected to show how this can be done and the results were compared to the outcome of the competition. We decided to make omnibus test for the results obtained with EI quality indicator on five objective problems. The reason is in the fact that, here, algorithm AGE-II outperformed the CVEA3, which was shown to be the best performing algorithm at the competition. After preparing the JSON input (accessible at http://cs.ijs.si/dl/dsctool/M05-ei-omnibus.json) the omnibus web service was executed. The obtained results indicate that the null hypothesis is rejected, since the calculated p-value is 1.09×10^{-10} and it is smaller than our previously set significance level (0.05). This means that there is a statistical significance in the data and post-hoc test should be performed to identify where these significant differences come from. In our case, the algorithm with the lowest mean value (AGE-II) was selected as the control algorithm and compared to the other algorithms. To apply the post-hoc test, the algorithm means obtained from the omnibus test were taken, the number of algorithms and the number of benchmark functions set (in our case 10 and 15, respectively), and the control algorithm specified (in our case AGE-II). Since the post-hoc statistic is dependent from the omnibus statistical test, the same statistical test must be applied in the post-hoc test (in our case, the Friedman test). After creating the JSON input (accessible at http://cs.ijs.si/dl/dsctool/M05-ei-posthoc.json), the posthoc web service was executed. The results show that algorithm AGE-II significantly outperforms every other algorithm. Therefore, what we have shown is that not only can the rankings change by applying a different quality indicator, but the differences between performances can be significant. If we would do a similar test on an HV quality indicator with five objective problems, the results would be reversed. Here, CVEA3 would be shown as an algorithm that significantly outperforms every other algorithm. However, if we would look at the results of any of the ensemble methods, we would see that there is no statistical significance between algorithms AGE-II and CVEA3, which in the end would be a general conclusion (assuming we would not search for the best algorithm according to some problem specific quality indicator).

With our experiments, we have shown how important it is to perform proper analysis since we were able to change outcomes of the study only by selecting different quality indicators and/or statistical approaches.

6. Conclusions

Performing a proper statistical analysis is an important task that must be taken with great care. This was clearly shown in Section 5, where we showed that the selection of quality indicators and the statistical approach that will be used to analyze this data has a big influence on the end results of the comparison. For these reasons, the DSCTool can help users to make proper statistical analysis quick and error free. It guides users through all analysis steps and provides them with all the required information that is needed to perform a proper statistical analysis and obtain final conclusions from their studies. There are still some decisions that should be made by the user such as a selection of significance level and choosing the relevant ensemble, but this requires much less knowledge than doing all of the statistics on their own.

Author Contributions: Conceptualization, P.K. and T.E.; methodology, T.E. and P.K.; software, P.K., and T.E.; validation, P.K., and T.E.; formal analysis, T.E. and P.K.; investigation, P.K. and T.E.; writing—original draft preparation, P.K.; writing—review and editing, T.E. and P.K.; funding acquisition, T.E. and P.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the financial support from the Slovenian Research Agency (research core funding No. P2-0098 and project No. Z2-1867) and from the European Union's Horizon 2020 research and innovation program under grant agreement No. 692286.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DSC	deep statistical comparison
EI	epsilon indicator
GD	generational distance
HV	hypervolume
IGD	inverse generational distance

References

1. Eftimov, T.; Korošec, P.; Seljak, B.K. Comparing multi-objective optimization algorithms using an ensemble of quality indicators with deep statistical comparison approach. In Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 27 November–1 December 2017; pp. 1–8.
2. Eftimov, T.; Korošec, P.; Seljak, B.K. A novel approach to statistical comparison of meta-heuristic stochastic optimization algorithms using deep statistics. *Inf. Sci.* **2017**, *417*, 186–215. [CrossRef]
3. Eftimov, T.; Korošec, P.; Seljak, B.K. Data-Driven Preference-Based Deep Statistical Ranking for Comparing Multi-objective Optimization Algorithms. In Proceedings of the International Conference on Bioinspired Methods and Their Applications, Paris, France, 16–18 May 2018; pp. 138–150.
4. Eftimov, T.; Petelin, G.; Korošec, P. DSCTool: A web-service-based framework for statistical comparison of stochastic optimization algorithms. *Appl. Soft Comput.* **2020**, *87*, 105977.10.1016/j.asoc.2019.105977. [CrossRef]
5. García, S.; Molina, D.; Lozano, M.; Herrera, F. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization. *J. Heuristics* **2009**, *15*, 617. [CrossRef]
6. Coello, C.A.C. Constraint-Handling Techniques Used with Evolutionary Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO'18)*; Association for Computing Machinery: New York, NY, USA, 2018; pp. 773–799.10.1145/3205651.3207855. [CrossRef]
7. Zitzler, E.; Thiele, L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.* **1999**, *3*, 257–271. [CrossRef]
8. Knowles, J.; Thiele, L.; Zitzler, E. A tutorial on the performance assessment of stochastic multiobjective optimizers. *Tik Rep.* **2006**, *214*, 327–332.
9. Van Veldhuizen, D.A.; Lamont, G.B. *Multiobjective Evolutionary Algorithm Research: A History and Analysis*; Technical Report; CiteSeer: Princeton, NJ, USA, 1998.
10. Riquelme, N.; Von Lücken, C.; Baran, B. Performance metrics in multi-objective optimization. In Proceedings of the 2015 Latin American Computing Conference (CLEI), Arequipa, Peru, 19–23 October 2015; pp. 1–11.
11. Eftimov, T.; Korošec, P.; Seljak, B.K. Disadvantages of statistical comparison of stochastic optimization algorithms. In Proceedings of the Bioinspired Optimizaiton Methods and their Applications (BIOMA), Bled, Slovenia, 18–20 May 2016; pp. 105–118.
12. Moore, D.S.; McCabe, G.P.; Craig, B. *Introduction to the Practice of Statistics*, 9th ed.; W. H. Freeman: New York City, NY, USA, 1998.
13. Eftimov, T.; Korošec, P.; KoroušićSeljak, B. Deep Statistical Comparison Applied on Quality Indicators to Compare Multi-objective Stochastic Optimization Algorithms. In *Machine Learning, Optimization, and Big Data*; Nicosia, G., Pardalos, P., Giuffrida, G., Umeton, R., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 76–87.

14. Eftimov, T.; Korosec, P.; Korousic-Seljak, B. The Behavior of Deep Statistical Comparison Approach for Different Criteria of Comparing Distributions. In Proceedings of the IJCCI, Funchal, Madeira, Portugal, 1–3 November 2017; pp. 73–82.
15. Eftimov, T.; Korošec, P.; Seljak, B.K. Deep statistical comparison applied on quality indicators to compare multi-objective stochastic optimization algorithms. In Proceedings of the International Workshop on Machine Learning, Optimization, and Big Data, Siena, Italy, 10–13 September 2017; pp. 76–87.
16. Boroushaki, S. Entropy-based weights for multicriteria spatial decision-making. *Yearb. Assoc. Pac. Coast Geogr.* **2017**, *79*, 168–187. [CrossRef]
17. Brans, J.P.; Mareschal, B. PROMETHEE methods. In *Multiple Criteria Decision Analysis: State of the Art Surveys*; Springer: New York, NY, USA, 2005; pp. 163–186.
18. Cheng, R.; Li, M.; Tian, Y.; Xiang, X.; Zhang, X.; Yang, S.; Jin, Y.; Yao, X. Competition on Many-Objective Optimization at 2018 IEEE Congress on Evolutionary Computation. 2020. Available online: https://www.cs.bham.ac.uk/~chengr/CEC_Comp_on_MaOO/2018/webpage.html (accessed on 2 April 2020).
19. Wagner, M.; Neumann, F. A Fast Approximation-Guided Evolutionary Multi-Objective Algorithm. In Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO'13), Amsterdam, The Netherlands, 6–10 July 2013; Association for Computing Machinery: New York, NY, USA, 2013; pp. 687–694.10.1145/2463372.2463448. [CrossRef]
20. Li, M.; Yang, S.; Liu, X. Pareto or Non-Pareto: Bi-Criterion Evolution in Multiobjective Optimization. *IEEE Trans. Evol. Comput.* **2016**, *20*, 645–665. [CrossRef]
21. Yuan, J.; Liu, H.; Gu, F. A Cost Value Based Evolutionary Many-Objective Optimization Algorithm with Neighbor Selection Strategy. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
22. Zhao, M.; Ge, H.; Han, H.; Sun, L. A Many-Objective Evolutionary Algorithm with Fast Clustering and Reference Point Redistribution. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–6.
23. Fritsche, G.; Pozo, A. A Hyper-Heuristic Collaborative Multi-objective Evolutionary Algorithm. In Proceedings of the 2018 7th Brazilian Conference on Intelligent Systems (BRACIS), Sao Paulo, Brazil, 22–25 October 2018; pp. 354–359.
24. Zhang, X.; Tian, Y.; Jin, Y. A Knee Point-Driven Evolutionary Algorithm for Many-Objective Optimization. *IEEE Trans. Evol. Comput.* **2015**, *19*, 761–776. [CrossRef]
25. Liu, Y.; Gong, D.; Sun, X.; Zhang, Y. Many-Objective Evolutionary Optimization Based on Reference Points. *Appl. Soft Comput.* **2017**, *50*, 344–355, doi:10.1016/j.asoc.2016.11.009. [CrossRef]
26. He, C.; Tian, Y.; Jin, Y.; Zhang, X.; Pan, L. A radial space division based evolutionary algorithm for many-objective optimization. *Appl. Soft Comput.* **2017**, *61*, 603–621.10.1016/j.asoc.2017.08.024. [CrossRef]
27. Cheng, R.; Jin, Y.; Olhofer, M.; Sendhoff, B. A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization. *IEEE Trans. Evol. Comput.* **2016**, *20*, 773–791. [CrossRef]
28. Cheng, R.; Li, M.; Tian, Y.; Zhang, X.; Yang, S.; Jin, Y.; Yao, X. A benchmark test suite for evolutionary many-objective optimization. *Complex Intell. Syst.* **2017**, *3*, 67–81. doi:10.1007/s40747-017-0039-7. [CrossRef]