Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Invited paper

A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms

Joaquín Derrac[a,*], Salvador García[b], Daniel Molina[c], Francisco Herrera[a]

[a] Department of Computer Science and Artificial Intelligence, CITIC-UGR (Research Center on Information and Communications Technology), University of Granada, 18071 Granada, Spain

[b] Department of Computer Science, University of Jaén, 23071 Jaén, Spain

[c] Department of Computer Engineering, University of Cadiz, 11003 Cadiz, Spain

ARTICLE INFO

Article history:
Received 18 October 2010
Received in revised form
22 December 2010
Accepted 8 February 2011
Available online 18 February 2011

Keywords:
Statistical analysis
Nonparametric statistics
Pairwise comparisons
Multiple comparisons
Evolutionary algorithms
Swarm intelligence algorithms

ABSTRACT

The interest in nonparametric statistical analysis has grown recently in the field of computational intelligence. In many experimental studies, the lack of the required properties for a proper application of parametric procedures – independence, normality, and homoscedasticity – yields to nonparametric ones the task of performing a rigorous comparison among algorithms.

In this paper, we will discuss the basics and give a survey of a complete set of nonparametric procedures developed to perform both pairwise and multiple comparisons, for multi-problem analysis. The test problems of the CEC'2005 special session on real parameter optimization will help to illustrate the use of the tests throughout this tutorial, analyzing the results of a set of well-known evolutionary and swarm intelligence algorithms. This tutorial is concluded with a compilation of considerations and recommendations, which will guide practitioners when using these tests to contrast their experimental results.

1. Introduction

In recent years, the use of statistical tests to improve the evaluation process of the performance of a new method has become a widespread technique in computational intelligence. Usually, they are employed inside the framework of any experimental analysis to decide when one algorithm is considered better than another. This task, which may not be trivial, has become necessary to confirm whether a new proposed method offers a significant improvement, or not, over the existing methods for a given problem.

Statistical procedures developed to perform statistical analyses can be categorized into two classes: parametric and nonparametric, depending on the concrete type of data employed [1]. Parametric tests have been commonly used in the analysis of experiments in computational intelligence. Unfortunately, they are based on assumptions which are most probably violated when analyzing the performance of stochastic algorithms based on computational intelligence [2,3]. These assumptions are known as independence, normality, and homoscedasticity. To overcome this problem, our interest is focused on nonparametric statistical procedures, which provide to the researcher a practical tool to use when the previous assumptions cannot be satisfied, especially in multi-problem analysis.

In this paper, the use of several nonparametric procedures for pairwise and multiple comparison procedures is illustrated. Our objectives are as follows.

- To give a comprehensive and useful tutorial about the use of nonparametric statistical tests in computational intelligence, using tests already proposed in several papers of the literature [2–5]. Through several examples of application, we will show their properties, and how the use of this complete framework can improve the way in which researchers and practitioners contrast the results achieved in their experimental studies.
- To analyze the lessons learned through their use, providing a wide list of guidelines which may guide users of these tests when selecting procedures for a given case of study.

For each kind of test, a complete case of application is shown. A contest held in the CEC'2005 special session on real parameter optimization defined a complete suite of benchmarking functions (publicly available; see [6]), considering several well-known domains for real parameter optimization. These benchmark functions will be used to compare several evolutionary and swarm intelligence continuous optimization techniques, whose differences will be contrasted through the use of nonparametric procedures.

[*] Corresponding author. Tel.: +34 958 240598; fax: +34 958 243317.
E-mail addresses: jderrac@decsai.ugr.es (J. Derrac), sglopez@ujaen.es (S. García), daniel.molina@uca.es (D. Molina), herrera@decsai.ugr.es (F. Herrera).

2210-6502/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.swevo.2011.02.002

To do so, this paper is organized as follows. Section 2 shows the experimental framework considered for the application of the statistical methods and gives some preliminary background. Section 3 describes the nonparametric tests for pairwise comparisons. Section 4 deals with multiple comparisons by designating a control method, whereas Section 5 deals with multiple comparisons among all methods. Section 6 surveys several recommendations and considerations on the use of nonparametric tests. Finally, Section 7 concludes this tutorial.

2. Preliminaries

In this section, the benchmark functions (Section 2.1) and the evolutionary and swarm intelligence algorithms considered for our case of study (Section 2.2) are presented. Furthermore, some basic concepts on inferential statistics are introduced (Section 2.3), providing the necessary background for properly presenting the statistical procedures included in this tutorial.

2.1. Benchmark functions: CEC'2005 special session on real parameter optimization

Thorough this paper, the results obtained in a experimental study regarding 9 well-known algorithms and 25 optimization functions will be used, illustrating the application of the different statistical methodologies considered. The nonparametric tests will be used to show significant statistical differences among the different algorithms of the study.

As benchmark suite, we have selected the 25 test problems of dimension 10 that appeared in the CEC'2005 special session on real parameter optimization [6]. This suite is composed of the following functions.

- 5 unimodal functions
  - F1: Shifted Sphere Function.
  - F2: Shifted Schwefel's Problem 1.2.
  - F3: Shifted Rotated High Conditioned Elliptic Function.
  - F4: Shifted Schwefel's Problem 1.2 with Noise in Fitness.
  - F5: Schwefel's Problem 2.6 with Global Optimum on Bounds.
- 20 multimodal functions
  - 7 basic functions.
    - F6: Shifted Rosenbrock's Function.
    - F7: Shifted Rotated Griewank Function without Bounds.
    - F8: Shifted Rotated Ackley's Function with Global Optimum on Bounds.
    - F9: Shifted Rastrigin's Function.
    - F10: Shifted Rotated Rastrigin's Function.
    - F11: Shifted Rotated Weierstrass Function.
    - F12: Schwefel's problem 2.13.
  - 2 expanded functions.
    - F13: Expanded Extended Griewank's plus Rosenbrock's Function (F8F2)
    - F14: Shifted Rotated Expanded Scaffers F6.
  - 11 hybrid functions. Each one (F15 to F25) has been defined through compositions of 10 out of the 14 previous functions (different in each case).

All functions are displaced in order to ensure that their optima can never be found in the center of the search space. In two functions, in addition, the optima cannot be found within the initialization range, and the domain of search is not limited (the optimum is out of the range of initialization).

2.2. Evolutionary and swarm intelligence algorithms

Our main case of study consist of the comparison of performance between 9 continuous optimization algorithms. Their main characteristics are described as follows.

- **PSO**: A classic Particle Swarm Optimization [7] model for numerical optimization has been considered. The parameters are $c1 = 2.8$, $c2 = 1.3$, and w from 0.9 to 0.4. Population is composed by 100 individuals.
- **IPOP-CMA-ES**: IPOP-CMA-ES is a restart Covariant Matrix Evolutionary Strategy (CMA-ES) with Increasing Population Size [8]. This CMA-ES variation detects premature convergence and launches a restart strategy that doubles the population size on each restart; by increasing the population size, the search characteristic becomes more global after each restart, which empowers the operation of the CMA-ES on multi-modal functions. For this algorithm, we have considered the default parameters. The initial solution is uniform randomly chosen from the domain, and the initial distribution size is a third of the domain size.
- **CHC**: The key idea of the CHC algorithm [9] concerns the combination of a selection strategy with a very high selective pressure and several components inducing a strong diversity. In [10], the original CHC model was extended to deal with real-coded chromosomes, maintaining its basis as much as possible. We have tested it using a real-parameter crossover operator, BLX-α (with $\alpha = 0.5$), and a population size of 50 chromosomes.
- **SSGA**: A real-coded Steady-State Genetic Algorithm specifically designed to promote high population diversity levels by means of the combination of the BLX-α crossover operator (with $\alpha = 0.5$) and the negative assortative mating strategy [11]. Diversity is favored as well by means of the BGA mutation operator [12].
- **SS-arit & SS-BLX**: Two instances of the classic Scatter Search model [13] have been included in the study: the original model with the arithmetical combination operator, and the same model using the BLX-α crossover operator (with $\alpha = 0.5$) [14].
- **DE-Exp & DE-Bin**: We have considered a classic Differential Evolution model [15], with no parameter adaptation. Two classic crossover operators proposed in the literature, *Rand/1/exp*, and *Rand/1/bin*, are applied. The *F* and *CR* parameters are fixed to 0.5 and 0.9, respectively, and the population size to 100 individuals.
- **SaDE**: Self-adaptive Differential Evolution [16] is a Differential Evolution model which can adapt its *CR* and *F* parameters for enhance its results. In this model, the population size has been fixed to 100 individuals.

All the algorithms have been run 50 times for each test function. Each run stops either when the error obtained is less than 10^{-8}, or when the maximal number of evaluations (100 000) is achieved. Table 1 shows the average error obtained for each one over the 25 benchmark functions considered.

2.3. Some basic concepts on inferential statistics

Single-problem and multi-problem analyses can usually be found contrasting the results of computational intelligence experiments, both in isolation [17] and simultaneously [18]. The first kind, single-problem analysis, deals with results obtained over several runs of the algorithms over a given problem, whereas multi-problem analysis considers a result per algorithm/problem pair.

Inside the field of inferential statistics, hypothesis testing [19] can be employed to draw inferences about one or more populations from given samples (results). In order to do that, two hypotheses, the null hypothesis H_0 and the alternative hypothesis H_1, are defined. The null hypothesis is a statement of no effect or no difference, whereas the alternative hypothesis represents the presence of an effect or a difference (in our case, significant differences between algorithms). When applying a statistical procedure to reject a hypothesis, a level of significance α is used to determine at which level the hypothesis may be rejected.

Table 1
Average error obtained in the 25 benchmark functions.

Function	PSO	IPOP-CMA-ES	CHC	SSGA	SS-BLX	SS-Arit	DE-Bin	DE-Exp	SaDE
F1	$1.234 \cdot 10^{-4}$	0.000	2.464	$8.420 \cdot 10^{-9}$	$3.402 \cdot 10$	1.064	$7.716 \cdot 10^{-9}$	$8.260 \cdot 10^{-9}$	$8.416 \cdot 10^{-9}$
F2	$2.595 \cdot 10^{-2}$	0.000	$1.180 \cdot 10^{2}$	$8.719 \cdot 10^{-5}$	1.730	5.282	$8.342 \cdot 10^{-9}$	$8.181 \cdot 10^{-9}$	$8.208 \cdot 10^{-9}$
F3	$5.174 \cdot 10^{4}$	0.000	$2.699 \cdot 10^{5}$	$7.948 \cdot 10^{4}$	$1.844 \cdot 10^{5}$	$2.535 \cdot 10^{5}$	$4.233 \cdot 10$	$9.935 \cdot 10$	$6.560 \cdot 10^{3}$
F4	2.488	$2.932 \cdot 10^{3}$	$9.190 \cdot 10$	$2.585 \cdot 10^{-3}$	6.228	5.755	$7.686 \cdot 10^{-9}$	$8.350 \cdot 10^{-9}$	$8.087 \cdot 10^{-9}$
F5	$4.095 \cdot 10^{2}$	$8.104 \cdot 10^{-10}$	$2.641 \cdot 10^{2}$	$1.343 \cdot 10^{2}$	2.185	$1.443 \cdot 10$	$8.608 \cdot 10^{-9}$	$8.514 \cdot 10^{-9}$	$8.640 \cdot 10^{-9}$
F6	$7.310 \cdot 10^{2}$	0.000	$1.416 \cdot 10^{6}$	6.171	$1.145 \cdot 10^{2}$	$4.945 \cdot 10^{2}$	$7.956 \cdot 10^{-9}$	$8.391 \cdot 10^{-9}$	$1.612 \cdot 10^{-2}$
F7	$2.678 \cdot 10$	$1.267 \cdot 10^{3}$	$1.269 \cdot 10^{3}$	$1.271 \cdot 10^{3}$	$1.966 \cdot 10^{3}$	$1.908 \cdot 10^{3}$	$1.266 \cdot 10^{3}$	$1.265 \cdot 10^{3}$	$1.263 \cdot 10^{3}$
F8	$2.043 \cdot 10$	$2.001 \cdot 10$	$2.034 \cdot 10$	$2.037 \cdot 10$	$2.035 \cdot 10$	$2.036 \cdot 10$	$2.033 \cdot 10$	$2.038 \cdot 10$	$2.032 \cdot 10$
F9	$1.438 \cdot 10$	$2.841 \cdot 10$	5.886	$7.286 \cdot 10^{-9}$	4.195	5.960	4.546	$8.151 \cdot 10^{-9}$	$8.330 \cdot 10^{-9}$
F10	$1.404 \cdot 10$	$2.327 \cdot 10$	7.123	$1.712 \cdot 10$	$1.239 \cdot 10$	$2.179 \cdot 10$	$1.228 \cdot 10$	$1.118 \cdot 10$	$1.548 \cdot 10$
F11	5.590	1.343	1.599	3.255	2.929	2.858	2.434	2.067	6.796
F12	$6.362 \cdot 10^{2}$	$2.127 \cdot 10^{2}$	$7.062 \cdot 10^{2}$	$2.794 \cdot 10^{2}$	$1.506 \cdot 10^{2}$	$2.411 \cdot 10^{2}$	$1.061 \cdot 10^{2}$	$6.309 \cdot 10$	$5.634 \cdot 10$
F13	1.503	1.134	$8.297 \cdot 10$	$6.713 \cdot 10$	$3.245 \cdot 10$	$5.479 \cdot 10$	1.573	$6.403 \cdot 10$	$7.070 \cdot 10$
F14	3.304	3.775	2.073	2.264	2.796	2.970	3.073	3.158	3.415
F15	$3.398 \cdot 10^{2}$	$1.934 \cdot 10^{2}$	$2.751 \cdot 10^{2}$	$2.920 \cdot 10^{2}$	$1.136 \cdot 10^{2}$	$1.288 \cdot 10^{2}$	$3.722 \cdot 10^{2}$	$2.940 \cdot 10^{2}$	$8.423 \cdot 10$
F16	$1.333 \cdot 10^{2}$	$1.170 \cdot 10^{2}$	$9.729 \cdot 10$	$1.053 \cdot 10^{2}$	$1.041 \cdot 10^{2}$	$1.134 \cdot 10^{2}$	$1.117 \cdot 10^{2}$	$1.125 \cdot 10^{2}$	$1.227 \cdot 10^{2}$
F17	$1.497 \cdot 10^{2}$	$3.389 \cdot 10^{2}$	$1.045 \cdot 10^{2}$	$1.185 \cdot 10^{2}$	$1.183 \cdot 10^{2}$	$1.279 \cdot 10^{2}$	$1.421 \cdot 10^{2}$	$1.312 \cdot 10^{2}$	$1.387 \cdot 10^{2}$
F18	$8.512 \cdot 10^{2}$	$5.570 \cdot 10^{2}$	$8.799 \cdot 10^{2}$	$8.063 \cdot 10^{2}$	$7.668 \cdot 10^{2}$	$6.578 \cdot 10^{2}$	$5.097 \cdot 10^{2}$	$4.482 \cdot 10^{2}$	$5.320 \cdot 10^{2}$
F19	$8.497 \cdot 10^{2}$	$5.292 \cdot 10^{2}$	$8.798 \cdot 10^{2}$	$8.899 \cdot 10^{2}$	$7.555 \cdot 10^{2}$	$7.010 \cdot 10^{2}$	$5.012 \cdot 10^{2}$	$4.341 \cdot 10^{2}$	$5.195 \cdot 10^{2}$
F20	$8.509 \cdot 10^{2}$	$5.264 \cdot 10^{2}$	$8.960 \cdot 10^{2}$	$8.893 \cdot 10^{2}$	$7.463 \cdot 10^{2}$	$6.411 \cdot 10^{2}$	$4.928 \cdot 10^{2}$	$4.188 \cdot 10^{2}$	$4.767 \cdot 10^{2}$
F21	$9.138 \cdot 10^{2}$	$4.420 \cdot 10^{2}$	$8.158 \cdot 10^{2}$	$8.522 \cdot 10^{2}$	$4.851 \cdot 10^{2}$	$5.005 \cdot 10^{2}$	$5.240 \cdot 10^{2}$	$5.420 \cdot 10^{2}$	$5.140 \cdot 10^{2}$
F22	$8.071 \cdot 10^{2}$	$7.647 \cdot 10^{2}$	$7.742 \cdot 10^{2}$	$7.519 \cdot 10^{2}$	$6.828 \cdot 10^{2}$	$6.941 \cdot 10^{2}$	$7.715 \cdot 10^{2}$	$7.720 \cdot 10^{2}$	$7.655 \cdot 10^{2}$
F23	$1.028 \cdot 10^{3}$	$8.539 \cdot 10^{2}$	$1.075 \cdot 10^{3}$	$1.004 \cdot 10^{3}$	$5.740 \cdot 10^{2}$	$5.828 \cdot 10^{2}$	$6.337 \cdot 10^{2}$	$5.824 \cdot 10^{2}$	$6.509 \cdot 10^{2}$
F24	$4.120 \cdot 10^{2}$	$6.101 \cdot 10^{2}$	$2.959 \cdot 10^{2}$	$2.360 \cdot 10^{2}$	$2.513 \cdot 10^{2}$	$2.011 \cdot 10^{2}$	$2.060 \cdot 10^{2}$	$2.020 \cdot 10^{2}$	$2.000 \cdot 10^{2}$
F25	$5.099 \cdot 10^{2}$	$1.818 \cdot 10^{3}$	$1.764 \cdot 10^{3}$	$1.747 \cdot 10^{3}$	$1.794 \cdot 10^{3}$	$1.804 \cdot 10^{3}$	$1.744 \cdot 10^{3}$	$1.742 \cdot 10^{3}$	$1.738 \cdot 10^{3}$

Instead of stipulating a priori a level of significance α, it is possible to compute the smallest level of significance that results in the rejection of H_0. This is the definition of the p-value, which is the probability of obtaining a result at least as extreme as the one that was actually observed, assuming that H_0 is true. It is a useful and interesting datum for many consumers of statistical analysis. A p-value provides information about whether a statistical hypothesis test is significant or not, and it also indicates something about *how significant* the result is: the smaller the p-value, the stronger the evidence against H_0. Most importantly, it does this without committing to a particular level of significance [20].

Parametric tests have been commonly used in the analysis of experiments in computational intelligence. For example, a common way to test whether the difference between the results of two algorithms is non-random is to compute a paired t-test, which checks whether the average difference in their performance over the problems is significantly different from zero. When comparing a set of multiple algorithms, the common statistical method for testing the differences between more than two related sample means is the repeated-measures ANOVA (or within-subjects ANOVA) [21].

Nonparametric tests, besides their original definition for dealing with nominal or ordinal data, can be also applied to continuous data by conducting ranking-based transformations, adjusting the input data to the test requirements [20]. They can perform two classes of analysis: pairwise comparisons and multiple comparisons. Pairwise statistical procedures perform individual comparisons between two algorithms, obtaining in each application a p-value independent from another one. Therefore, in order to carry out a comparison which involves more than two algorithms, multiple comparisons tests should be used. In $1 \times N$ comparisons, a control method is highlighted (the best performing algorithm) through the application of the test. Then, all hypotheses of equality between the control method and the rest can be tested by the application of a set of post-hoc procedures. $N \times N$ comparisons, considering the hypotheses of equality between all existing pairs of algorithms, are also possible, with the inclusion of specific post-hoc procedures for this task.

Table 2
Nonparametric statistical procedures considered in this tutorial.

Type of comparison	Procedures	Section
Pairwise comparisons	Sign test	3.1
	Wilcoxon test	3.2
Multiple comparisons ($1 \times N$)	Multiple sign test	4.1
	Friedman test	4.2
	Friedman Aligned ranks	4.2
	Quade test	4.2
	Contrast Estimation	4.4
Multiple comparisons ($N \times N$)	Friedman test	5

Table 3
Associated post-hoc procedures.

Type of comparison	Procedures	Section
Multiple comparisons ($1 \times N$)	Bonferroni	4.3
	Holm	4.3
	Hochberg	4.3
	Hommel	4.3
	Holland	4.3
	Rom	4.3
	Finner	4.3
	Li	4.3
Multiple comparisons ($N \times N$)	Nemenyi	5
	Holm	5
	Shaffer	5
	Bergmann	5

In this tutorial, we describe the use of several pairwise and multiple comparison procedures. Tables 2 and 3 enumerates the statistical tests and the post-hoc procedures considered, respectively. Furthermore, we present here some common notation that is used.

- n is the number of problems considered. i is its associated index.
- k is the number of algorithms included in the comparison. j is its associated index.
- d denotes the difference of performance between two algorithms in a given problem.

This notation will be employed throughout the study, unless a particular case is stated explicitly.

Table 4
Critical values for the two-tailed sign test at $\alpha = 0.05$ and $\alpha = 0.1$. An algorithm is significantly better than another if it performs better on at least the cases presented in each row.

#Cases	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
$\alpha = 0.05$	5	6	7	7	8	9	9	10	10	11	12	12	13	13	14	15	15	16	17	18	18
$\alpha = 0.1$	5	6	6	7	7	8	9	9	10	10	11	12	12	13	13	14	14	15	16	16	17

Table 5
Example of Sign test for pairwise comparisons. SaDE shows a significant improvement over PSO, CHC, and SSGA, with a level of significance $\alpha = 0.05$, and over SS-Arit, with a level of significance $\alpha = 0.1$.

SaDE	PSO	IPOP-CMA-ES	CHC	SSGA	SS-BLX	SS-Arit	DE-Bin	DE-Exp
Wins (+)	20	15	20	18	16	17	13	9
Loses (−)	5	10	5	7	9	8	12	16
Detected differences	$\alpha = 0.05$	–	$\alpha = 0.05$	$\alpha = 0.05$	–	$\alpha = 0.1$	–	–

3. Pairwise comparisons

Pairwise comparisons are the simplest kind of statistical tests that a researcher can apply within the framework of an experimental study. Such tests are directed to compare the performance of two algorithms when applied to a common set of problems. In multi-problem analysis, a value for each pair algorithm/problem is required (often an average value from several runs).

In this section, first we focus our attention on a quick and easy, yet not very powerful, procedure, which can provide a first snapshot about the comparison: the Sign test (Section 3.1). Then, we will introduce the use of the Wilcoxon signed ranks test (Section 3.2), as a example of a simple, yet safe and robust, nonparametric test for pairwise statistical comparisons. Examples thorough this section will focus in characterizing the behavior of SaDE, in 1×1 comparisons with the rest of algorithms considered.

3.1. A simple first-sight procedure: the Sign test

A popular way to compare the overall performances of algorithms is to count the number of cases on which an algorithm is the overall winner. Some authors also use these counts in inferential statistics, with a form of two-tailed binomial test that is known as the Sign test [22]. If both algorithms compared are, as assumed under the null hypothesis, equivalent, each should win on approximately $n/2$ out of n problems.

The number of wins is distributed according to a binomial distribution; for a greater number of cases, the number of wins is under the null hypothesis distributed according to $n\left(n/2, \sqrt{n}/2\right)$, which allows for the use of the z-test: if the number of wins is at least $n/2 + 1.96 \cdot \sqrt{n}/2$ (or, for a quick rule of a thumb, $n/2 + \sqrt{n}$), then the algorithm is significantly better with $p < 0.05$.

Table 4 shows the critical number of wins needed to achieve both $\alpha = 0.05$ and $\alpha = 0.1$ levels of significance. Note that, since tied matches support the null hypothesis, they should not be discounted when applying this test, but split evenly between the two algorithms; if there is an odd number of them, one should be ignored.

Example 1. In our experimental framework, performing a Sign test to compare the results of SaDE is quite simple. It only requires counting the number of wins achieved either by SaDE or by the comparison algorithm. Then, using Table 4, we can highlight those cases where a significant difference is detected. Table 5 summarizes this process.

3.2. The Wilcoxon signed ranks test

The Wilcoxon signed ranks test is used for answering the following question: do two samples represent two different populations? It is a nonparametric procedure employed in hypothesis testing situations, involving a design with two samples. This is analogous to the paired t-test in nonparametric statistical procedures; therefore, it is a pairwise test that aims to detect significant differences between two sample means, that is, the behavior of two algorithms.

Wilcoxon's test is defined as follows. Let d_i be the difference between the performance scores of the two algorithms on ith out of n problems (if these performance scores are known to be represented in different ranges, they can be normalized to the interval [0, 1], in order to not prioritize any problem; see [23]). The differences are ranked according to their absolute values; in case of ties, the practitioner can apply one of the available methods existing in the literature [24] (ignore ties, assign the highest rank, compute all the possible assignments and average the results obtained in every application of the test, and so on), although we recommend the use of average ranks for dealing with ties (for example, if two differences are tied in the assignation of ranks 1 and 2, assign rank 1.5 to both differences).

Let R^+ be the sum of ranks for the problems in which the first algorithm outperformed the second, and R^- the sum of ranks for the opposite. Ranks of $d_i = 0$ are split evenly among the sums; if there is an odd number of them, one is ignored:

$$R^+ = \sum_{d_i>0} \operatorname{rank}(d_i) + \frac{1}{2}\sum_{d_i=0} \operatorname{rank}(d_i)$$

$$R^- = \sum_{d_i<0} \operatorname{rank}(d_i) + \frac{1}{2}\sum_{d_i=0} \operatorname{rank}(d_i). \qquad (1)$$

Let T be the smaller of the sums, $T = \min(R^+, R^-)$. If T is less than or equal to the value of the distribution of Wilcoxon for n degrees of freedom ([25], Table B.12), the null hypothesis of equality of means is rejected; this will mean that a given algorithm outperforms the other one, with the p-value associated. Given its widespread use, the computation of the p-value for this test is usually included in well-known statistical software packages (SPSS, SAS, R, etc.).

The Wilcoxon signed ranks test is more sensitive than the t-test. It assumes commensurability of differences, but only qualitatively: greater differences still count for more, which is probably desired, but the absolute magnitudes are ignored. From the statistical point of view, the test is safer since it does not assume normal distributions. Also, the outliers (exceptionally good/bad performances of a few problems) have less effect on the Wilcoxon test than on the t-test. The Wilcoxon test assumes continuous differences d_i; therefore, they should not be rounded to one or two decimals, since this would decrease the power of the test in the case of a high number of ties.

Table 6
Wilcoxon signed ranks test results. SaDE shows an improvement over PSO, CHC, and SSGA, with a level of significance $\alpha = 0.01$, over IPOP-CMA-ES and SS-Arit, with $\alpha = 0.05$, and over SS-BLX, with $\alpha = 0.1$.

Comparison	R^+	R^-	p-value	Comparison	R^+	R^-	p-value
SaDE versus PSO	261	64	0.00673	SaDE versus SS-BLX	232	93	0.06262
SaDE versus IPOP-CMA-ES	239	86	0.03934	SaDE versus SS-Arit	243	82	0.02958
SaDE versus CHC	287	38	0.00038	SaDE versus DE-Bin	176	149	>0.2
SaDE versus SSGA	260	65	0.00737	SaDE versus DE-Exp	119	206	>0.2

Example 2. When using Wilcoxon's test in our experimental study, the first step is to compute the R^+ and R^- related to the comparisons between SaDE and the rest of algorithms. Once they have been obtained, their associated p-values can be computed. Note that, for every comparison, the property $R^+ + R^- = n \cdot (n+1)/2$ must be true.

Table 6 shows the R^+, R^-, and p-values computed for all the pairwise comparisons concerning SaDE (the p-values have been computed by using SPSS). As the table states, SaDE shows a significant improvement over PSO, CHC, and SSGA, with a level of significance $\alpha = 0.01$, over IPOP-CMA-ES and SS-Arit, with $\alpha = 0.05$, and over SS-BLX, with $\alpha = 0.1$.

4. Multiple comparisons with a control method

One of the most frequent situations where the use of statistical procedures is requested is in the joint analysis of the results achieved by various algorithms. The groups of differences between these methods (also called blocks) are usually associated with the problems met in the experimental study. For example, in a multiple problem comparison, each block corresponds to the results offered over a specific problem. When referring to multiple comparisons tests, a block is composed of three or more subjects or results, each one corresponding to the performance evaluation of the algorithm over the problem.

In pairwise analysis, if we try to extract a conclusion involving more than one pairwise comparison, we will obtain an accumulated error coming from its combination. In statistical terms, we are losing the control on the Family-Wise Error Rate (FWER), defined as the probability of making one or more false discoveries among all the hypotheses when performing multiple pairwise tests. The true statistical significance for combining pairwise comparisons is given by

$$\begin{aligned} p &= P(Reject H_0|H_0 true) \\ &= 1 - P(Accept H_0|H_0 true) \\ &= 1 - P(Accept A_k = A_i, i = 1, \ldots, k-1|H_0 true) \\ &= 1 - \prod_{i=1}^{k-1} P(Accept A_k = A_i|H_0 true) \\ &= 1 - \prod_{i=1}^{k-1} [1 - P(Reject A_k = A_i|H_0 true)] \\ &= 1 - \prod_{i=1}^{k-1} (1 - p_{H_i}). \end{aligned}$$

Therefore, a pairwise comparison test, such as Wilcoxon's test, should not be used to conduct various comparisons involving a set of algorithms, because the FWER is not controlled.

This section is devoted to describing the use of several procedures for multiple comparisons considering a control method. In this sense, a control method can be defined as the most interesting algorithm for the researcher of the experimental study (usually its new proposal). Therefore, its performance will be contrasted against the rest of algorithms of the study.

The contents of this section are summarized as follows.

- First, we will introduce the use of the Sign test for multiple comparisons. This Multiple Sign test (Section 4.1) is a not very powerful method for detecting significant differences between algorithms, but it is still a quick and easy procedure which can be interesting for a first glance at the results.
- The best-known procedure for testing the differences between more than two related samples, the Friedman test, will be introduced in Section 4.2. In that section, we will also include the use of its extension, the Iman–Davenport test, and two advanced versions: the Friedman Aligned Ranks test and the Quade test.
- In Section 4.3, we will illustrate the use of a family of post-hoc procedures, as a suitable complement for the Friedman-related tests. Given a control method and the ranks of the Friedman (or any related) test, these post-hoc methods allow us to determine which algorithms are significantly better/worse than it.
- Finally, in Section 4.4, we present a procedure to estimate the differences between several algorithms: the Contrast Estimation of medians. This method is very recommendable if we assume that the global performance is reflected by the magnitudes of the differences among the performances of the algorithms.

4.1. Multiple Sign test

Given a control labeled algorithm, the Sign test for multiple comparisons allows us to highlight those ones whose performances are statistically different when compared with the control algorithm. This procedure, proposed in [26,27], proceeds as follows.

1. Represent by $x_{i,1}$ and x_{ij} the performances of the control and the jth algorithm in the ith problem.
2. Compute the signed differences $d_{i,j} = x_{i,j} - x_{i,1}$. That is, pair each performance with the control and, in each problem, subtract the control performance from the performance of the jth algorithm.
3. Let r_j equal the number of differences, $d_{i,j}$, that have the less frequently occurring sign (either positive or negative) within a pairing of an algorithm with the control.
4. Let M_1 be the median response of a sample of results of the control algorithm and M_j be the median response of a sample of results of the jth algorithm. Apply one of the following decision rules.
   - For testing H_0: $M_j \geq M_1$ against H_1: $M_j < M_1$, reject H_0 if the number of minus signs is less than or equal to the critical value of R_j appearing in Table A.21 in Appendix A for $k-1$ (number of algorithms excluding control), n (number of problems), and the chosen experimentwise error rate.
   - For testing H_0: $M_j \leq M_1$ against H_1: $M_j > M_1$, reject H_0 if the number of plus signs is less than or equal to the critical value of R_j appearing in Table A.21 in Appendix A for $k-1$, n, and the chosen experimentwise error rate.

Example 3. Labeling SaDE as our control algorithm, we may reuse the results shown in Table 5 for applying the Multiple Sign test. Suppose we choose a level of significance $\alpha = 0.05$ and let our hypotheses be H_0: $M_j \geq M_1$ and H_1: $M_j < M_1$; that is, our control algorithm SaDE is significantly better than the remaining algorithms. Reference to Table A.21 for $m = 8$ ($m = k - 1$) and $n = 25$ reveals that the critical value of R_j is 5. Since the number of minuses in the pairwise comparison between the control and PSO and CHC is equal to 5, we may conclude that SaDE has a significantly better performance than them. However, the null hypothesis cannot be rejected in the pairwise comparison among the rest of the comparison algorithms, so we cannot highlight more significant differences using this test.

Note the differences between the results of the single Sign test (see Example 1) and the Multiple Sign test. Although the former states that PSO, CHC, SSGA, and SS-Arit are statistically improved by SaDE, the latter only detects significant differences between PSO and CHC when compared with SaDE. This result is caused by the control of the FWER, which prevents the rejection of the null hypothesis of equality for SSGA and SS-Arit, in contrast with the single pairwise comparison performed in the Example 1.

In fact, it is possible to argue that, if we reduce the number of algorithms in the comparison to six ($m = 5$), excluding three algorithms from the study, we would detect significant differences between SaDE and SSGA ($\alpha = 0.1$), due to the critical value of the test being increased to 7. However, this would lead to assuming that significant differences found are only valid in the presence of the six algorithms considered, and not in the presence of the full set of nine algorithms of the comparison. Note that this means that the rejection of pairwise hypotheses with a control algorithm is influenced by the rest of methods considered in the comparison, if the Multiple Sign test is used.

4.2. The Friedman, Friedman Aligned Ranks, and Quade tests

The Friedman test [28,29] (Friedman two-way analysis of variances by ranks) is a nonparametric analog of the parametric two-way analysis of variance. It can be used for answering the following question: in a set of k samples (where $k \geq 2$), do at least two of the samples represent populations with different median values?. The Friedman test is the analog of the repeated measures ANOVA in nonparametric statistical procedures; therefore, it is a multiple comparisons test that aims to detect significant differences between the behavior of two or more algorithms.

The null hypothesis for Friedman's test states equality of medians between the populations. The alternative hypothesis is defined as the negation of the null hypothesis, so it is non-directional.

The first step in calculating the test statistic is to convert the original results to ranks. They are computed using the following procedure.

1. Gather observed results for each algorithm/problem pair.
2. For each problem i, rank values from 1 (best result) to k (worst result). Denote these ranks as r_i^j $(1 \leq j \leq k)$.
3. For each algorithm j, average the ranks obtained in all problems to obtain the final rank $R_j = \frac{1}{n}\sum_i r_i^j$.

Thus, it ranks the algorithms for each problem separately; the best performing algorithm should have the rank of 1, the second best rank 2, etc. Again, in case of ties, we recommend computing average ranks. Under the null hypothesis, which states that all the algorithms behave similarly (therefore their ranks R_j should be equal) the Friedman statistic F_f can be computed as

$$F_f = \frac{12n}{k(k+1)}\left[\Sigma_j R_j^2 - \frac{k(k+1)^2}{4}\right], \tag{2}$$

which is distributed according to a χ^2 distribution with $k - 1$ degrees of freedom, when n and k are big enough (as a rule of a thumb, $n > 10$ and $k > 5$). For a smaller number of algorithms and problems, exact critical values have been computed [22,25].

Iman and Davenport [30] proposed a derivation from the Friedman statistic given that this last metric often produces a conservative effect not desired. The proposed statistic is

$$F_{ID} = \frac{(n-1)\chi_F^2}{n(k-1) - \chi_F^2} \tag{3}$$

which is distributed according to an F distribution with $k - 1$ and $(k-1)(N-1)$ degrees of freedom. See Table A10 in [22] to find the critical values.

A drawback of the ranking scheme employed by the Friedman test is that it allows for intra-set comparisons only. When the number of algorithms for comparison is small, this may pose a disadvantage, since inter-set comparisons may not be meaningful. In such cases, comparability among problems is desirable.

In the method of aligned ranks [31] for the Friedman test, a value of location is computed as the average performance achieved by all algorithms in each problem. Then, the difference between the performance obtained by an algorithm and the value of location is obtained. This step is repeated for each combination of algorithms and problems.

The resulting differences (aligned observations), which keep their identities with respect to the problem and the combination of algorithms to which they belong, are then ranked from 1 to $k \cdot n$ relative to each other. This ranking scheme is the same as that employed by a multiple comparison procedure which employs independent samples, such as the Kruskal–Wallis test [32]. The ranks assigned to the aligned observations are called aligned ranks.

The Friedman Aligned Ranks test statistic can be defined as

$$F_{AR} = \frac{(k-1)\left[\sum_{j=1}^{k}\hat{R}_j^2 - (kn^2/4)(kn+1)^2\right]}{\{[kn(kn+1)(2kn+1)]/6\} - (1/k)\sum_{i=1}^{n}\hat{R}_i^2}, \tag{4}$$

where $\hat{R}_i$ is equal to the rank total of the ith problem and $\hat{R}_j$ is the rank total of the jth algorithm.

The test statistic F_{AR} is compared for significance with a χ^2 distribution with $k - 1$ degrees of freedom. Critical values can be found in Table A3 in [22].

Finally, we will introduce a last test for performing multiple comparisons: the Quade test [33]. This test, in contrast to Friedman's, takes into account the fact that some problems are more difficult or that the differences registered on the run of various algorithms over them are larger (the Friedman test considers all problems to be equal in terms of importance). Therefore, the rankings computed on each problem could be scaled depending on the differences observed in the algorithms' performances, obtaining, as a result, a weighted ranking analysis of the sample of results.

The procedure starts by finding the ranks r_i^j in the same way as the Friedman test does. The next step requires the original values of performance of the algorithms x_i^j. Ranks are assigned to the problems themselves according to the size of the sample range in each problem. The sample range within problems i is the difference between the largest and the smallest observations within that problem:

$$\text{Range in problem: } i = \max_j x_i^j - \min_j x_i^j. \tag{5}$$

Obviously, there are n sample ranges, one for each problem. Assign rank 1 to the problem with the smallest range, rank 2 to the second smallest, and so on to the problem with the largest range, which gets rank n. Use average ranks in the case of ties.

Let $Q_1, Q_2, \ldots, Q_N$ be the ranks assigned to problems $1, 2, \ldots, N$, respectively.

Finally, the problem rank Q_i is multiplied by the difference between the rank within problem i, r_i^j, and the average rank within problems, $(k+1)/2$, to get the product S_i^j, where

$$S_i^j = Q_i\left[r_i^j - \frac{k+1}{2}\right] \tag{6}$$

is a statistic that represents the relative size of each observation within the problem, adjusted to reflect the relative significance of the problem in which it appears. Also, we may define S_j as the sum for each algorithm, $S_j = \sum_{i=1}^{n} S_i^j$, for $j = 1, 2, \ldots, k$.

For convenience, and to establish a relationship with the Friedman test, we will also use rankings without average adjusting,

$$W_i^j = Q_i\left[r_i^j\right], \tag{7}$$

and the average ranking for the jth algorithm, T_j, given as

$$T_j = \frac{W_j}{n(n+1)/2}, \tag{8}$$

where $W_j = \sum_{i=1}^{n} W_i^j$, for $j = 1, 2, \ldots, k$.

Some definitions must be made for computing the test statistic, F_Q. Let the terms A and B be

$$A = n(n+1)(2n+1)k(k+1)(k-1)/72 \tag{9}$$

$$B = \frac{1}{n}\sum_{j=1} kS_j^2. \tag{10}$$

The test statistic, F_Q, is

$$F_Q = \frac{(n-1)B}{A-B}, \tag{11}$$

which is distributed according to the F-distribution with $k-1$ and $(k-1)(n-1)$ degrees of freedom (the critical values can be found in Table A10 in [22]). When computing the statistic, note that, if $A = B$, we must consider the point to be in the critical region of the statistical distribution.

For each of these tests (Friedman, Iman–Davenport, Friedman Aligned Ranks, or Quade tests), once the proper statistics have been computed, it is possible to compute a p-value through normal approximations [34] (in the Quade test, if $A = B$, the p-value is computed as $(1/k!)^{n-1}$). If the existence of significant differences is found (that is, the null hypothesis is rejected), we can proceed with a post-hoc procedure to characterize these differences (see Section 4.3). A Java package developed to compute the rankings for these test, the CONTROLTEST package, can be obtained at the SCI2S thematic public website *Statistical Inference in Computational Intelligence and Data Mining*.[1]

Table 7
Error rates achieved (Example 4).

Error	A	B	C	D
P1	2.711	3.147	2.515	2.612
P2	7.832	9.828	7.832	7.921
P3	0.012	0.532	0.122	0.005
P4	3.431	4.111	3.401	3.401

Table 8
Friedman ranks (Example 4).

Friedman	A	B	C	D
P1	3	4	1	2
P2	1.5	4	1.5	3
P3	2	4	3	1
P4	3	4	1.5	1.5
Average	2.375	4	1.250	1.875

Table 9
Friedman Aligned ranks (Example 4).

Friedman Aligned	A	B	C	D
P1	12	14	4	10
P2	1.5	16	1.5	3
P3	8	13	11	7
P4	9	15	5.5	5.5
Average	7.625	14.5	5.5	6.375

Table 10
Quade ranks (Example 4).

Quade	A	B	C	D
P1	1 (6)	3 (8)	−3 (2)	−1 (4)
P2	−4 (6)	6 (16)	−4 (6)	2 (12)
P3	−0.5 (2)	1.5 (4)	0.5 (3)	−1.5 (1)
P4	1.5 (9)	4.5 (12)	−3 (4.5)	−3 (4.5)
S_j	−2	15	−9.5	−3.5
T_j	2.3	4	1.55	2.15

Example 4. To understand the computation of the ranks for the Friedman, Friedman Aligned and Quade procedures, we firstly present a toy example, considering the error rates achieved by four algorithms (labeled from A to D) over four problems (labeled from P1 to P4). Table 7 shows them.

Table 8 depicts the ranks computed through the Friedman test. As can be seen in the table, C is the best performing algorithm of our example, whereas B is the worst.

Table 9 depicts the ranks computed through the Friedman Aligned test. In that table we may see how aligned observations modify the way in which ranks are computed, increasing greatly, for example, the rank of Algorithm B over problem P2, or decreasing the rank of Algorithm C over problem P1.

Finally, Table 10 depicts the ranks computed through the Quade test, considering both weighted ranks Si^j and ranks without weighting Wi^j (between brackets). From this table, we may highlight the differences between the importance assigned to each problem. For example, ranks assigned to P2 are greater than the rest (in terms of absolute value), whereas ranks assigned to P3 are significantly less (which can be interpreted as considering problem P2 as hard, and problem P3 as easy).

Although the order between algorithms given by the three procedures is the same, it is interesting to see how the different procedures allow us to distinguish some problems from the rest, following a given criterion.

Example 5. Continuing with our experimental study, the ranks of the Friedman, Friedman Aligned, and Quade tests can be computed for all the algorithms considered, following the guidelines exposed in this section. Table 11 shows them, highlighting DE-Exp as the best performing algorithm of the comparison, with a rank of 3.5, 84.74, and 3.1123 for the Friedman, Friedman Aligned, and Quade tests, respectively.

The p-values computed through the statistics of each of the tests considered (0.000018, 0.006357, and $1.20327 \cdot 10^{-07}$) and the Iman–Davenport extension ($F_f = 5.267817$, p-value: 0.000006) strongly suggest the existence of significant differences among the algorithms considered.

[1] http://sci2s.ugr.es/sicidm/.

Table 11
Ranks achieved by the Friedman, Friedman Aligned, and Quade tests in the main case of study. DE-Exp achieves the best rank in the three procedures. The statistics computed and related p-values are also shown.

Algorithms	Friedman	Friedman Aligned	Quade
PSO	7	138.84	6.5415
IPOP-CMA-ES	4.84	116.12	4.7415
CHC	6.28	157.4	7.1785
SSGA	5.5	129.14	5.8769
SS-BLX	4.64	107.92	5.1108
SS-Arit	5.4	107.8	5.6123
DE-Bin	4	88.28	3.5538
DE-Exp	3.5	84.74	3.1123
SaDE	3.84	86.76	3.2723
Statistic	35.99733	21.31479	6.63067
p-value	0.000018	0.006357	$1.20327 \cdot 10^{-07}$

4.3. Post-hoc procedures

The main drawback of the Friedman, Iman–Davenport, Friedman Aligned, and Quade tests is that they only can detect significant differences over the whole multiple comparison, being unable to establish proper comparisons between some of the algorithms considered. When the aim of the application of the multiple tests is to perform a comparison considering a control method and a set of algorithms, a family of hypotheses can be defined, all related to the control method. Then, the application of a post-hoc test can lead to obtaining a p-value which determines the degree of rejection of each hypothesis.

A family of hypotheses is a set of logically interrelated hypotheses of comparisons which, in $1 \times N$ comparisons, compares the $k-1$ algorithms of the study (excluding the control) with the control method, whereas in $N \times N$ comparisons, it considers the $k(k-1)/2$ possible comparisons among algorithms. Therefore, the family will be composed of $k-1$ or $k(k-1)/2$ hypotheses, respectively, which can be ordered by its p-value, from lowest to highest.

The p-value of every hypothesis in the family can be obtained through the conversion of the rankings computed by each test by using a normal approximation. The test statistic for comparing the ith algorithm and jth algorithm, z, depends on the main nonparametric procedure used.

- **Friedman test:**

$$z = (R_i - R_j) / \sqrt{\frac{k(k+1)}{6n}}, \tag{12}$$

where R_i and R_j are the average rankings by the Friedman test of the algorithms compared [35].

- **Friedman Aligned test:**

$$z = (\hat{R}_i - \hat{R}_j) / \sqrt{\frac{k(n+1)}{6}}, \tag{13}$$

where $\hat{R}_i$ and $\hat{R}_j$ are the average rankings by the Friedman Aligned Ranks test of the algorithms compared [35,32].

- **Quade test:**

$$z = (T_i - T_j) / \sqrt{\frac{k(k+1)(2n+1)(k-1)}{18n(n+1)}}, \tag{14}$$

where $T_i = \frac{W_i}{n(n+1)/2}$, $T_j = \frac{W_j}{n(n+1)/2}$ and W_i and W_i are the rankings without average adjusting by the Quade test of the algorithms compared [22].

Table 12
Ranks obtained in Example 6.

Ranks	Friedman	Aligned	Quade
IPOP-CMA-ES	2.48	51.96	2.3785
CHC	3.12	65.92	3.4185
SS-BLX	2.44	48.52	2.48
SaDE	1.96	35.6	1.7231

Table 13
Friedman z-values and p-values (Example 6).

Friedman	z	Unadjusted p-value
CHC	3.176791	0.001489
IPOP-CMA-ES	1.424079	0.154424
SS-BLX	1.314534	0.188667

Table 14
Friedman Aligned z-values and p-values (Example 6).

Friedman Aligned	z	Unadjusted p-value
CHC	3.694997	0.000220
IPOP-CMA-ES	1.993739	0.046181
SS-BLX	1.574517	0.115368

Table 15
Quade z-values and p-values (Example 6).

Quade	z	Unadjusted p-value
CHC	3.315129	0.000916
SS-BLX	1.480076	0.138853
IPOP-CMA-ES	1.281529	0.200008

Example 6. To better illustrate the practical differences between the three tests and their respective approximations for obtaining the p-value of every hypothesis (which are called unadjusted p-values; see below), we will consider here a short example, where ranks (Table 12), z-values, and unadjusted p-values (Tables 13–15) are computed for four algorithms: IPOP-CMA-ES, CHC, BLX and SaDE.

Several differences can be highlighted: the Friedman Aligned test shows a higher power than Friedman test (the unadjusted p-values obtained by the former are substantially lower, especially in the SaDE versus IPOP-CMA-ES case). Comparing the Friedman test with the Quade test, it can be seen that the latter considers differences between SaDE and SS-BLX significantly greater than those between SaDE and IPOP-CMA-ES. In this sense, the Quade test is supporting the fact that IPOP-CMA-ES is achieving better results in harder problems than SS-BLX, when both are compared considering SaDE as the control method.

However, these p-values are not suitable for multiple comparisons. When a p-value is considered in a multiple test, it reflects the probability error of a certain comparison, but it does not take into account the remaining comparisons belonging to the family. If k algorithms are being compared and in each comparison the level of significance is α, then in a single comparison the probability of not making a Type I error (rejecting a true null hypothesis) is $(1-\alpha)$, and the probability of not making a Type I error in the $k-1$ comparison is $(1-\alpha)^{(k-1)}$. Therefore, the probability of making one or more Type I error is $1-(1-\alpha)^{(k-1)}$. For instance, if $\alpha = 0.05$ and $k = 9$ this is 0.33, which is rather high.

Adjusted p-values (APVs) can deal with this problem. Since they take into account the family error accumulated, multiple tests can be conducted without disregarding the FWER. Moreover, APVs can be compared directly with any chosen significance level α. Therefore, their use is recommended since they provide more information in a statistical analysis.

The z-value in all cases is used to find the corresponding probability (p-value) from the table of normal distribution $N(0, 1)$, which is then compared with an appropriate level of significance α (Table A1 in [22]). The post-hoc tests differ in the way they adjust the value of α to compensate for multiple comparisons.

Next, we will define a set of post-hoc procedures and we will explain how to compute the APVs depending on the post-hoc procedure used in the analysis, following the indications given in [36]. The notation used for describing the computation of the APVs has the following differences (compared with the notation used in the rest of the paper).

- Indexes i and j each correspond to a concrete comparison or hypothesis in the family of hypotheses, according to an incremental order of their p-values. Index i always refers to the hypothesis in question whose APV is being computed and index j refers to another hypothesis in the family.
- p_j is the p-value obtained for the jth hypothesis.

The procedures of p-value adjustment can be classified into several classes.

- **one-step**:
  - The Bonferroni–Dunn procedure (Dunn–Sidak approximation) [37]: this adjusts the value of α in a single step by dividing it by the number of comparisons performed, $(k-1)$. This procedure is the simplest but it also has little power.

    Bonferroni APV_i : $\min\{v, 1\}$, where $v = (k-1)p_i$.
- **step-down**:
  - The Holm procedure [38]: this adjusts the value of α in a step-down manner. Let $p_1, p_2, \ldots, p_{k-1}$ be the ordered p-values (smallest to largest), so that $p_1 \leq p_2 \leq \cdots \leq p_{k-1}$, and let $H_1, H_2, \ldots, H_{k-1}$ be the corresponding hypotheses. The Holm procedure rejects H_1 to H_{i-1} if i is the smallest integer such that $p_i > \alpha/(k-1)$. Holm's step-down procedure starts with the most significant p-value. If p_1 is below $\alpha/(k-1)$, the corresponding hypothesis is rejected and we are allowed to compare p_2 with $\alpha/(k-2)$. If the second hypothesis is rejected, the test proceeds with the third, and so on. As soon as a certain null hypothesis cannot be rejected, all the remaining hypotheses are retained as well.

    Holm APV_i : $\min\{v, 1\}$, where $v = \max\{(k-j)p_j : 1 \leq j \leq i\}$.
  - The Holland procedure [39]: this also adjusts the value of α in a step-down manner, as Holm's method. It rejects H_1 to H_{i-1} if i is the smallest integer so that $p_i > 1-(1-\alpha)^{k-i}$.

    Holland APV_i : $\min\{v, 1\}$, where $v = \max\{1-(1-p_j)^{(k-j)} : 1 \leq j \leq i\}$.
  - The Finner procedure [40]: this also adjusts the value of α in a step-down manner, as Holm's and Holland's methods do. It rejects H_1 to H_{i-1} if i is the smallest integer so that $p_i > 1-(1-\alpha)^{(k-1)/i}$.

    Finner APV_i : $\min\{v, 1\}$, where $v = \max\{1-(1-p_j)^{(k-1)/j} : 1 \leq j \leq i\}$.
- **step-up**:
  - The Hochberg procedure [41] adjusts the value of α in a step-up way. It works by comparing the largest p-value with α, the next largest with $\alpha/2$, the next with $\alpha/3$, and so forth until it finds a hypothesis it can reject. All hypotheses with smaller p-values are then rejected as well.

    Hochberg APV_i : $\max\{(k-j)p_j : (k-1) \geq j \geq i\}$.
  - The Hommel procedure [42], which is more complicated than the rest, works by finding the largest j for which $p_{n-j+k} > k\alpha/j$ for all $k = 1, \ldots, j$. If no such j exists, it rejects all hypotheses; otherwise, it rejects all for which $p_i \leq \alpha/j$.

    Hommel APV_i: see Hommel's APV algorithm (Fig. 1).

```
Set $APV_i = p_i$ for all $i$
foreach $j = k-1, k-2, \ldots, 2$ (in that order) do
    Let B = $\emptyset$
    foreach $i, i > (k-1-j)$ do
        Compute value $c_i = (j \cdot p_i)/(j+i-k+1)$
        $B = B \cup c_i$
    Find the smallest $c_i$ value in $B$; call it $c_{min}$
    if $APV_i < c_{min}$ then
        $APV_i = c_{min}$
    foreach $i, i \leq (k-1-j)$ do
        Let $c_i = min(c_{min}, j \cdot p_i)$
        if $APV_i < c_i$ then
            $APV_i = c_i$
```

Fig. 1. Method for computing Hommel's test APV.

  - The Rom procedure [43]: Rom developed a modification to Hochberg's procedure to increase its power. It works in exactly the same way as the Hochberg procedure, except that the α-values are computed through the expression

$$\alpha_{k-i} = \left[\sum_{j=1}^{i-1} \alpha^j - \sum_{j=1}^{i-2} \binom{i}{k} \alpha_{k-1-j}^{i-j} \right] / i, \qquad (15)$$

    where $\alpha_{k-1} = \alpha$ and $\alpha_{k-2} = \alpha/2$.

    Rom APV_i : $\max\{(r_{k-j})p_j : (k-1) \geq j \geq i\}$, where r_{k-j} can be obtained from Eq. (15) ($r = \{1, 2, 3, 3.814, 4.755, 5.705, 6.655, \ldots\}$).
- **two-step**:
  - The Li procedure [44]: Li proposed a two-step rejection procedure.
    - ∗ Step 1: Reject all H_i if $p_{k-1} \leq \alpha$. Otherwise, accept the hypothesis associated to p_{k-1} and go to Step 2.
    - ∗ Step 2: Reject any remaining H_i with $p_i \leq (1-p_{k-1})/(1-\alpha)\alpha$. Li APV_i: $p_i/(p_i + 1 - p_{k-1})$.

The CONTROLTEST package, available at the SCI2S thematic public website *Statistical Inference in Computational Intelligence and Data Mining*, also contains an implementation of all the post-hoc tests (see footnote 1).

Example 7. By following the indications given for the eight post-hoc procedures considered, Tables 16–18 show the p-values obtained, using the ranks computed by the Friedman, Friedman Aligned, and Quade tests, respectively.

As we can see in the tables, the Friedman test shows a significant improvement of DE-Exp over PSO, CHC, SSGA, and SS-Arit for all the post-hoc procedures considered, except for the Bonferroni–Dunn one. The Finner and Li tests exhibit the most powerful behavior, reaching the lowest p-values in the comparisons.

The Friedman Aligned test only confirms the improvement of DE-Exp over PSO, CHC, and SSGA for every post-hoc procedure considered, except Bonferroni–Dunn and Li, which fail to highlight the differences between DE-Exp and SSGA as significant. The Finner and Rom procedures show the most powerful behavior in this category.

Finally, the Quade test does not find any significant difference between DE-Exp and the rest of algorithms. This result support the conclusion that, although DE-Exp obtains better results than the *weaker* algorithms of our experimental study (PSO, CHC, and so on), these behave similarly or better in the most difficult problems, and thus performance differences are not detected if the relative difficulties of the problems are taken into account.

Table 16
Adjusted p-values for the Friedman test (DE-Exp is the control method).

Friedman	Unadjusted	Bonferroni	Holm	Hochberg	Hommel	Holland	Rom	Finner	Li
PSO	0.000006	0.000050	0.000050	0.000050	0.000050	0.000050	0.000047	0.000050	0.000018
CHC	0.000332	0.002656	0.002324	0.002324	0.002324	0.002322	0.002210	0.001327	0.000978
SSGA	0.009823	0.078586	0.058940	0.058940	0.049116	0.057511	0.056042	0.025981	0.028137
SS-Arit	0.014171	0.113371	0.070857	0.070857	0.070857	0.068877	0.067384	0.028142	0.040093
IPOP-CMA-ES	0.083642	0.669139	0.334569	0.334569	0.282186	0.294885	0.319017	0.130431	0.197766
SS-BLX	0.141093	1.0	0.423278	0.423278	0.423278	0.366366	0.423278	0.183552	0.293707
DE-Bin	0.518605	1.0	1.0	0.660706	0.660706	0.768259	0.660706	0.566345	0.604506
SaDE	0.660706	1.0	1.0	0.660706	0.660706	0.768259	0.660706	0.660706	0.660706

Table 17
Adjusted p-values for the Friedman Aligned test (DE-Exp is the control method).

Friedman Aligned	Unadjusted	Bonferroni	Holm	Hochberg	Hommel	Holland	Rom	Finner	Li
CHC	0.000079	0.000635	0.000635	0.000635	0.000635	0.000635	0.000604	0.000635	0.000907
PSO	0.003300	0.026401	0.023101	0.023101	0.023101	0.022873	0.021963	0.013135	0.036400
SSGA	0.015888	0.127104	0.095328	0.095328	0.095328	0.091621	0.090642	0.041809	0.153880
IPOP-CMA-ES	0.088320	0.706559	0.441599	0.441599	0.353280	0.370186	0.419957	0.168839	0.502727
SS-BLX	0.208043	1.0	0.832172	0.631221	0.624129	0.606625	0.631221	0.311471	0.704264
SS-Arit	0.210407	1.0	0.832172	0.631221	0.631221	0.606625	0.631221	0.311471	0.706612
DE-Bin	0.847534	1.0	1.0	0.912638	0.912638	0.976754	0.912638	0.883457	0.906555
SaDE	0.912638	1.0	1.0	0.912638	0.912638	0.976754	0.912638	0.912638	0.912638

Table 18
Adjusted p-values for the Quade test (DE-Exp is the control method).

Quade	Unadjusted	Bonferroni	Holm	Hochberg	Hommel	Holland	Rom	Finner	Li
CHC	0.021720	0.173762	0.173762	0.173762	0.173762	0.161111	0.165195	0.161111	0.231846
PSO	0.052904	0.423235	0.370330	0.370330	0.369115	0.316471	0.352093	0.195409	0.423683
SSGA	0.118631	0.949049	0.711787	0.711787	0.593156	0.531245	0.676797	0.285908	0.622427
SS-Arit	0.158192	1.0	0.790962	0.790962	0.632769	0.577269	0.752197	0.29136	0.687327
SS-BLX	0.259289	1.0	1.0	0.928037	0.777867	0.69898	0.928037	0.38136	0.782754
IPOP-CMA-ES	0.357754	1.0	1.0	0.928037	0.928037	0.735086	0.928037	0.445882	0.832533
DE-Bin	0.803179	1.0	1.0	0.928037	0.928037	0.961261	0.928037	0.843964	0.917769
SaDE	0.928037	1.0	1.0	0.928037	0.928037	0.961261	0.928037	0.928037	0.928037

4.4. Contrast Estimation

Contrast Estimation based on medians [45,46] can be used to estimate the difference between the performance of two algorithms. It assumes that the expected differences between performances of algorithms are the same across problems. Therefore, the performance of algorithms is reflected by the magnitudes of the differences between them in each domain.

The interest of this test lies in estimating the contrast between medians of samples of results considering all pairwise comparisons. The test obtains a quantitative difference computed through medians between two algorithms over multiple problems, proceeding as follows.

1. For every pair of k algorithms in the experiment, compute the difference between the performances of the two algorithms in each of the n problems. That is, compute the differences

$$D_{i(u,v)} = x_{iu} - x_{iv}, \tag{16}$$

where $i = 1, \ldots, n$; $u = 1, \ldots, k$; $v = 1, \ldots, k$. (Consider only performance pairs where $u < v$.)

2. Find the median of each set of differences (Z_{uv}, which can be regarded as the *unadjusted estimator* of the medians of the algorithms u and v, $M_u - M_v$). Since $Z_{uv} = Z_{vu}$, it is only required to compute Z_{uv} in those cases where $u < v$. Also, note that $Z_{uu} = 0$.
3. Compute the mean of each set of unadjusted medians having the same first subscript, m_u:

$$m_u = \frac{\sum_{j=1}^{k} Z_{uj}}{k}, \quad u = 1, \ldots, k. \tag{17}$$

4. The estimator of $M_u - M_v$ is $m_u - m_v$, where u and v range from 1 through k. For example, the difference between M_1 and M_2 is estimated by $m_1 - m_2$.

These estimators can be understood as an advanced global performance measure. Although this test cannot provide a probability of error associated with the rejection of the null hypothesis of equality, it is especially useful to estimate by how far an algorithm outperforms another one.

An implementation of the Contrast Estimation procedure can be found in the CONTROLTEST package, which can be obtained at the SCI2S thematic public website *Statistical Inference in Computational Intelligence and Data Mining* (see footnote 1).

Example 8. In our experimental analysis, we can compute the set of estimators of medians directly from the average error results. Table 19 shows the estimations computed for each algorithm.

Focusing our attention in the rows of the table, we may highlight the performance of SaDE (all its related estimators are negative;, that is, it achieves very low error rates considering median estimators) and the Scatter Search-based approaches; on the other hand, CHC and PSO achieve higher error rates in our experimental study.

Table 19
Contrast estimation results. The estimators highlight SaDE, SS-BLX, and SS-Arit as the best performing algorithms.

Estimation	PSO	IPOP-CMA-ES	CHC	SSGA	SS-BLX	SS-Arit	DE-Bin	DE-Exp	SaDE
PSO	0	11.172	−23.671	10.495	24.010	21.150	15.115	17.631	25.035
IPOP-CMA-ES	−11.172	0	−34.843	−0.677	12.838	9.978	3.943	6.459	13.863
CHC	23.671	34.843	0	34.166	47.681	44.821	38.786	41.302	48.706
SSGA	−10.495	0.677	−34.166	0	13.514	10.655	4.620	7.136	14.539
SS-BLX	−24.010	−12.838	−47.681	−13.514	0	−2.859	−8.895	−6.378	1.025
SS-Arit	−21.150	−9.978	−44.821	−10.655	2.859	0	−6.036	−3.519	3.884
DE-Bin	−15.115	−3.943	−38.786	−4.620	8.895	6.036	0	2.516	9.920
DE-Exp	−17.631	−6.459	−41.302	−7.136	6.378	3.519	−2.516	0	7.403
SaDE	−25.035	−13.863	−48.706	−14.539	−1.025	−3.884	−9.920	−7.403	0

5. Multiple comparisons among all methods

Friedman's test is an omnibus test which can be used to carry out these types of comparison. It allows us to detect differences considering the global set of algorithms. Once Friedman's test rejects the null hypothesis, we can proceed with a post-hoc test in order to find the concrete pairwise comparisons which produce differences. In the previous section, we focused on procedures that control the FWER when comparing with a control algorithm, arguing that the objective of a study is to test whether a newly proposed algorithm is better than the existing ones. For this reason, we have described and studied procedures such as the Bonferroni–Dunn, Holm and Hochberg methods.

When our interest lies in carrying out a multiple comparison in which all possible pairwise comparisons need to be computed ($N \times N$ comparison), two classic procedures that can be used are the Holm test (the same as was described in Section 4.3) and the Nemenyi procedure [47]. This procedure adjusts the value of α in a single step by dividing it by the number of comparisons performed, $m = k(k-1)/2$. It is the simplest of this family, but it also has little power.

The hypotheses being tested belonging to a family of all pairwise comparisons are logically interrelated; thus not all combinations of true and false hypotheses are possible. As a simple example of such a situation, suppose that we want to test the three hypotheses of pairwise equality associated with the pairwise comparisons of three algorithms $M_i, i = 1, 2, 3$. It is easily seen from the relations among the hypotheses that, if any one of them is false, at least one other must be false. For example, if M_1 is better/worse than M_2, then it is not possible that M_1 has the same performance as M_3 and M_2 has the same performance as M_3. M_3 must be better/worse than M_1 or M_2 or the two algorithms at the same time. Thus, there cannot be one false and two true hypotheses among these three.

Based on this argument, Shaffer proposed two procedures which make use of the logical relation among the family of hypotheses for adjusting the value of α [48].

- Shaffer's static procedure: following Holm's step-down method, at stage j, instead of rejecting H_i if $p_i \leq \alpha/(m-i+1)$, reject H_i if $p_i \leq \alpha/t_i$, where t_i is the maximum number of hypotheses which can be true given that any $(i, \ldots, 1)$ hypotheses are false. It is a static procedure; that is, $t_1, \ldots, t_m$ are fully determined for the given hypotheses $H_1, \ldots, H_m$, independent of the observed p-values. The possible numbers of true hypotheses, and thus the values of t_1 can be obtained from the recursive formula

$$S(k) = \bigcup_{j=1}^{k} \left\{ \binom{j}{2} + x : x \in S(k-j) \right\}, \tag{18}$$

where $S(k)$ is the set of possible numbers of true hypotheses with k algorithms being compared, $k \geq 2$, and $S(0) = S(1) = \{0\}$.

```
Input: $C = \{c_1, c_2, \ldots, c_k\}$:list of classifiers
Let $E = \emptyset$
$E = E \cup \{$set of all possible and distinct pairwise comparisons using $C\}$
if $E == \emptyset$ then
    return $E$
foreach possible divisions of $C$ into two subsets $C_1$ and $C_2$, $c_k \in C_2$ and $C_1 \neq \emptyset$ do
    $E_1$ =obtainExhaustive($C_1$)
    $E_2$ =obtainExhaustive($C_2$)
    $E = E \cup E_1$
    $E = E \cup E_2$
    foreach family of hypotheses $e_1$ of $E_1$ do
        foreach family of hypotheses $e_2$ of $E_2$ do
            $E = E \cup (e_1 \cup e_2)$
return $E$
```

Fig. 2. obtainExhaustive(C). Algorithm for obtaining all exhaustive sets in Bergmann's procedure.

- Shaffer's dynamic procedure: this increases the power of the first by substituting $\alpha = t_i$ at stage i by the value $\alpha = t_i^*$, where t_i^* is the maximum number of hypotheses that could be true, given that the previous hypotheses are false. It is a dynamic procedure, since t_i^* depends not only on the logical structure of the hypotheses, but also on the hypotheses already rejected at step i. Obviously, this procedure has more power than the first one. However, we will not use this second procedure, given that it is included in an advanced procedure which we will describe in the following.

In [49], a procedure was proposed based on the idea of finding all elementary hypotheses which cannot be rejected. In order to formulate Bergmann–Hommel's procedure, we need the following definition.

Definition 1. An index set of hypotheses $I \subseteq \{1, \ldots, m\}$ is called exhaustive if exactly all $H_j, j \in I$, could be true.

Under this definition, the Bergmann–Hommel procedure works as follows.

- Bergmann and Hommel procedure: reject all H_j with $j \notin A$, where the acceptance set A, given as

$$A = \bigcup \{I : I \text{ exhaustive }, min\,\{P_i : i \in I\} > \alpha/\,|I|\}, \tag{19}$$

is the index set of null hypotheses which are retained.

For this procedure, one has to check for each subset I of $\{1, \ldots, m\}$ if I is exhaustive, which leads to intensive computation. Due to this fact, we will obtain a set, named E, which will contain all the possible exhaustive sets of hypotheses for a certain comparison. A rapid algorithm which was described in [50] allows a substantial reduction in computing time. Once the E set is obtained, the hypotheses that do not belong to the A set are rejected.

Fig. 2 shows a valid algorithm for obtaining all the exhaustive sets of hypotheses, using as input a list of algorithms C. E is a set of families of hypotheses; likewise, a family of hypotheses is a set of hypotheses. The most important step in the algorithm is step 6. It performs a division of the algorithms into two subsets, in which the last algorithm k always is inserted in the second subset and the first subset cannot be empty. In this way, we ensure that a subset yielded in a division is never empty and no repetitions are produced.

Finally, we will explain how to compute the APVs for the three post-hoc procedures described above, following the indications given in [51].

- Nemenyi $APV_i : \min\{v : 1\}$, where $v = m \cdot p_i$.
- Holm APV_i (using it in all pairwise comparisons): $\min\{v : 1\}$, where $v = \max\{(m - j + 1)p_j : 1 \leq j \leq i\}$.
- Shaffer static $APV_i : \min\{v : 1\}$, where $v = \max\{t_j p_j : 1 \leq j \leq i\}$.
- Bergmann–Hommel $APV_i : \min\{v : 1\}$, where $v = \max\{\|I\| \cdot min\{p_j, j \in I\} : I \text{ exhaustive} ; i \in I\}$.

where m is the number of possible comparisons in an all pairwise comparisons design; that is, $m = \frac{k \cdot (k-1)}{2}$.

An implementation of the Friedman Test for multiple comparisons, with all its related post-hoc procedures, can be found in the MULTIPLETEST package, which can be obtained at the SCI2S thematic public website *Statistical Inference in Computational Intelligence and Data Mining* (see footnote 1).

Example 9. Starting from the analysis performed by the Friedman test over our experimental results (see Example 7), we can raise the 36 hypotheses of equality among the 9 algorithms of our study, and apply the above-mentioned methods to contrast them. Table 20 lists all the hypotheses and the p-values achieved.

Using a level of significance $\alpha = 0.1$, only six hypotheses are rejected by the Nemenyi method. These hypotheses show the improvement of DE-Exp and SaDE over PSO and CHC, and DE-Bin and SS-BLX over PSO. The Holm and Shaffer methods reject an additional hypothesis, thus confirming the improvement of DE-Bin over CHC. Finally, the Bergmann procedure rejects eight hypotheses, the last one being the equality between PSO and IPOP-CMA-ES. None of the remaining 28 hypotheses can be rejected using these procedures.

6. Considerations and recommendations on the use of nonparametric tests

This section notes some considerations and recommendations concerning the nonparametric tests presented in this tutorial. Their characteristics as well as suggestions on some of their aspects and details of the multiple comparisons tests are presented. With this aim, some general considerations and recommendations are given first (Section 6.1). Then, some advanced guidelines for multiple comparisons with a control method (Section 6.2) and multiple comparisons among all methods (Section 6.3) are provided.

6.1. General considerations

- By using nonparametric statistical procedures, it is possible to analyze any unary performance measure (that is, associated to a single algorithm) with a defined range. This range does not have to be limited; thus, comparisons considering running times, memory requirements, and so on, are feasible.
- Being able to be applied in multi-domain comparisons, nonparametric statistical procedures can compare both deterministic and stochastic algorithms simultaneously, providing that their results are represented as a sample for each pair of algorithm/domain.
- For the application of these methods, only a result for each pair of algorithm/domain is required. A known and standardized procedure must be followed to gather them, using average results from several executions when considering stochastic algorithms.
- An appropriate number of algorithms in contrast with an appropriate number of case problems are needed to be used in order to employ each type of test. The number of algorithms used in multiple comparisons procedures must be lower than the number of case problems. The previous statement may not be true for the Wilcoxon test. The influence of the number of case problems used is more noticeable in multiple comparison procedures than in Wilcoxon's test [2,3].
- Although Wilcoxon's test and the post-hoc tests for multiple comparisons are nonparametric statistical tests, they operate in a different way. The main difference lies in the computation of the ranking. Wilcoxon's test computes a ranking based on differences between case problems independently, whereas the Friedman test and its derivative procedures compute the ranking between algorithms [2,3].
- In relation to the sample size (number of case problems when performing Wilcoxon's or Friedman's tests in a multi-problem analysis), there are two main aspects to be determined. First, the minimum sample size considered acceptable for each test needs to be stipulated. There is no established agreement about this specification. Statisticians have studied the minimum sample size when a certain power of the statistical test is expected. In our case, the employment of a sample size as large as possible is preferable because the power of the statistical tests (defined as the probability that the test will reject a false null hypothesis) will increase. Moreover, in a multi-problem analysis, the increase of the sample size depends on the availability of new case problems (which should be well known in computational intelligence or data mining field). Second, we have to study how the results are expected to vary if there were a larger sample size available. In all statistical tests used for comparing two or more samples, an increase of the sample size benefits the power of the test. In the following items, we will state that Wilcoxon's test is less influenced by this factor than Friedman's test. Finally, as a rule of thumb, the number of case problems in a study should be $n = a \cdot k$, where $a \geq 2$ [2,3].
- Although there is not a theoretical maximum number of domains to use in a comparison, it can be derived from the central limit theorem that, if this number is too high, the results may be unreliable. If the number of domains grows too much, statistical tests can lose credibility, as they may start highlighting true insignificant hypotheses as significant ones. For the Wilcoxon's test, a maximum of 30 domains is suggested [4]. For multiple comparisons, a value of $n \geq 8 \cdot k$ could be too high, obtaining no significant comparisons as a result [2,3].
- Taking into account the previous observation and knowing the operations performed by the nonparametric tests, we can deduce that Wilcoxon's test is influenced by the number of case problems used. On the other hand, both the number of algorithms and case problems are crucial when we refer to multiple comparisons tests (such as Friedman's test), given that all the critical values depend on the value of n (see the expressions above). However, the increasing/decreasing of the number of case problems rarely affects the computation of the ranking. In these procedures, the number of functions used is an important factor to be considered when we want to control the FWER [2,3].
- Another interesting procedure considered in this paper is related to Contrast Estimation based on medians between two samples of results. Contrast Estimation in nonparametric statistics is used for computing the real differences between two algorithms, considering the median measure the most important.

Table 20
Adjusted *p*-values for tests for multiple comparisons among all methods.

i	Hypothesis	Unadjusted *p*	Nemenyi	Holm	Shaffer	Bergmann
1	PSO versus DE-Exp	0.000006	0.000224	0.000224	0.000224	0.000224
2	PSO versus SaDE	0.000045	0.001624	0.001579	0.001263	0.001263
3	PSO versus DE-Bin	0.000108	0.00387	0.003655	0.00301	0.002365
4	CHC versus DE-Exp	0.000332	0.011952	0.010956	0.009296	0.009296
5	CHC versus SaDE	0.001633	0.058772	0.052242	0.045712	0.034284
6	PSO versus SS-BLX	0.002313	0.08328	0.071713	0.064773	0.04164
7	CHC versus DE-Bin	0.003246	0.116841	0.097367	0.090876	0.051929
8	PSO versus IPOP-CMA-ES	0.005294	0.190602	0.15354	0.148246	0.095301
9	SSGA versus DE-Exp	0.009823	0.353638	0.275052	0.275052	0.216112
10	SS-Arit versus DE-Exp	0.014171	0.51017	0.382627	0.311771	0.255085
11	SSGA versus SaDE	0.032109	1.0	0.834835	0.706398	0.513744
12	CHC versus SS-BLX	0.03424	1.0	0.856006	0.753286	0.513744
13	PSO versus SS-Arit	0.038867	1.0	1.0	0.855076	0.621874
14	SS-Arit versus SaDE	0.044015	1.0	1.0	0.968322	0.621874
15	SSGA versus DE-Bin	0.052808	1.0	1.0	1.0	0.63369
16	PSO versus SSGA	0.052808	1.0	1.0	1.0	0.686498
17	IPOP-CMA-ES versus CHC	0.063023	1.0	1.0	1.0	0.756271
18	SS-Arit versus DE-Bin	0.070701	1.0	1.0	1.0	0.756271
19	IPOP-CMA-ES versus DE-Exp	0.083642	1.0	1.0	1.0	1.0
20	SS-BLX versus DE-Exp	0.141093	1.0	1.0	1.0	1.0
21	IPOP-CMA-ES versus SaDE	0.196706	1.0	1.0	1.0	1.0
22	CHC versus SS-Arit	0.255925	1.0	1.0	1.0	1.0
23	SSGA versus SS-BLX	0.266889	1.0	1.0	1.0	1.0
24	IPOP-CMA-ES versus DE-Bin	0.278172	1.0	1.0	1.0	1.0
25	SS-BLX versus SaDE	0.3017	1.0	1.0	1.0	1.0
26	CHC versus SSGA	0.313946	1.0	1.0	1.0	1.0
27	SS-BLX versus SS-Arit	0.326516	1.0	1.0	1.0	1.0
28	PSO versus CHC	0.352622	1.0	1.0	1.0	1.0
29	IPOP-CMA-ES versus SSGA	0.394183	1.0	1.0	1.0	1.0
30	SS-BLX versus DE-Bin	0.40867	1.0	1.0	1.0	1.0
31	IPOP-CMA-ES versus SS-Arit	0.469706	1.0	1.0	1.0	1.0
32	DE-Bin versus DE-Exp	0.518605	1.0	1.0	1.0	1.0
33	DE-Exp versus SaDE	0.660706	1.0	1.0	1.0	1.0
34	IPOP-CMA-ES versus SS-BLX	0.796253	1.0	1.0	1.0	1.0
35	DE-Bin versus SaDE	0.836354	1.0	1.0	1.0	1.0
36	SSGA versus SS-Arit	0.897279	1.0	1.0	1.0	1.0

Taking into account that the samples of results in computational intelligence experiments rarely fulfill the needed conditions for a safe use of parametric tests, the computation of nonparametric contrast estimation through the use of medians is very useful. For example, one could provide, apart from the average values of accuracies over various problems reported by the methods compared, the contrast estimation between them over multiple problems, which is a safer metric in multi-problem environments [46].

- Finally, we want to remark that the choice of any of the statistical procedures presented in this paper for conducting an experimental analysis should be justified by the researcher. The use of the most powerful procedures does not imply that the results obtained by a given proposal will be better. The choice of a statistical technique is ruled by a trade-off between its power and its complexity when it comes to being used or explained to non-expert readers in statistics [46].

6.2. *Multiple comparisons with a control method*

- A multiple comparison of various algorithms must be carried out first by using a statistical method for testing the differences among the related samples means, that is, the results obtained by each algorithm. Once this test rejects the hypothesis of equivalence of means, the detection of the concrete differences among the algorithms can be done with the application of post-hoc statistical procedures, which are methods used for comparing a control algorithm with two or more algorithms [2,3].
- An appropriate number of algorithms in contrast with an appropriate number of case problems are needed to be used in order to employ each type of test. The number of algorithms used in multiple comparisons procedures must be lower than the number of case problems. In general, *p*-values are lower on increasing the number of case problems used in multiple comparison procedures (so long as this number does not exceed $n \geq 8 \cdot k$); therefore, the differences among the algorithms are more detectable [2,3].
- As we have suggested, multiple comparisons tests must be used when we want to establish a statistical comparison of the results reported among various algorithms. We focus on cases when a method is compared against a set of algorithms. It could be carried out first by using a statistical method for testing the differences among the related samples means, that is, the results obtained by each algorithm. There are three alternatives: the Friedman test with the Iman–Davenport extension, the Friedman Aligned Ranks test, and the Quade test. Once one of these tests rejects the hypothesis of equivalence of medians, the detection of the specific differences among the algorithms can be made with the application of post-hoc statistical procedures, which are methods used for specifically comparing a control algorithm with two or more algorithms [46].
- In this kind of test, it is possible to use just the rankings obtained when establishing a classification between the algorithms, and even employ them to measure their performance differences. However, this cannot be used to conclude that a given proposal outperform the rest, unless the null hypothesis is rejected.
- Although, by definition, post-hoc statistical procedures can be applied in an independent way from the rejection of the null hypothesis, it is advisable to check this rejection firstly.

- Holm's procedure can always be considered better than Bonferroni–Dunn's procedure, because it appropriately controls the FWER and it is more powerful than Bonferroni–Dunn's procedure. We strongly recommend the use of Holm's method in a rigorous comparison. Nevertheless, the results offered by the Bonferroni–Dunn test are suitable to be visualized in graphical representations [2,3].
- Hochberg's procedure is more powerful than Holm's procedure. The differences between it and Holm's procedure are in practice rather small. We recommend the use of this test together with Holm's method [2,3].
- An alternative to directly performing a comparison between a control algorithm and a set of algorithms is the Multiple Sign test. It has been described in this paper, and an example of its use has been provided. We have shown that this procedure is rapid and easy to apply, but it has low power with respect to more advanced techniques. We recommend its use when the differences reported by the control method with respect to the rest of algorithms are very clear for a certain performance metric [46].
- Apart from the well-known Friedman test, we can use two alternatives which differ in the ranking computation. Both the Friedman Aligned Rank test and the Quade test can be used under the same circumstances as the Friedman test. The differences in power between Friedman Aligned Ranks test and the Quade test are unknown, but we encourage the use of these tests when the number of algorithms to be compared is low [46].
- As we have described, the Quade test adds to the ranking computation of Friedman's test a weight factor computed through the maximum and minimum differences in a problem. This implies that those algorithms that obtain further positive results in diverse problems could benefit from this test. The use of this test should be regulated, because it is very sensitive to the choice of problems. If a researcher decided to include a subgroup of an already studied group of problems where in most of them the proposal obtained good results, this test would report excessive significant differences. On the other hand, for specific problems in which we are interested in quantifying the real differences obtained between algorithms, the use of this test can be justified. We recommend the use of this procedure under justified circumstances and with special caution [46].
- In relation to the post-hoc procedures shown, the differences of power between the methods are rather small, with some exceptions. The Bonferroni–Dunn test should not be used in spite of its simplicity, because it is a very conservative test and many differences may not be detected. Five procedures (those of Holm, Hochberg, Hommel, Holland, and Rom) have a similar power. Although the Hommel and Rom procedures are the two most powerful procedures, they also are the most difficult to be applied and to be understood. A good alternative is to use the Finner test, which is easy to comprehend and offers better results than the remaining tests, except the Li test in some cases [46].
- The Li test is even simpler than the Finner, Holm, or Hochberg tests. This test needs to check only two steps and to know the greatest unadjusted p-value in the comparison, which is easy to obtain. The author declares that the power of his test is highly influenced by the p-value of the last hypothesis of the family and, when it is lower than 0.5, the test will be more powerful than the rest of post-hoc methods. However, we recommend that it be used with care and only when the differences between the control algorithm and the rest seem to be high in the performance measure analyzed [46].

6.3. Multiple comparisons among all methods

- When comparing all algorithms among themselves, we do not recommend the use of Nemenyi's test, because it is a very conservative procedure, and many of the obvious differences may not be detected [5].
- However, conducting the Shaffer static procedure means a not very significant increase of the difficulty with respect to the Holm procedure. Moreover, the benefit of using information about logically related hypothesis is noticeable; thus we strongly encourage the use of this procedure [5].
- Bergmann–Hommel's procedure is the best performing one, but it is also the most difficult to understand and is computationally expensive. We recommend its use when the situation requires it (that is, when the differences among the algorithms compared are not very significant), given that the results it obtains are as valid as using other testing procedures [5].

7. Conclusions

In this work, we have shown a complete set of nonparametric statistical procedures and their application to contrast the results obtained in experimental studies of continuous optimization algorithms. The wide set of methods considered, ranging from basic techniques such as the Sign test or Contrast Estimation, to more advanced approaches such as the Friedman Aligned and Quade tests, include tools which can help practitioners in many situations in which the results of an experimental study need to be contrasted.

For a better understanding, all the procedures described in this paper have been applied to a comprehensive case of study, analyzing the results of nine well-known evolutionary and swarm intelligence algorithms over the set of 25 benchmark functions considered in the CEC'2005 special session. This study has been extended with a list of considerations, in which we discuss some important issues concerning the behavior and applicability of these tests (and emphasize the use of the most appropriate test depending on the circumstances and type of comparison).

Finally, we encourage the use of nonparametric tests whenever there exists a necessity of analyzing results obtained by evolutionary or swarm intelligence algorithms for continuous optimization problems in multi-problem analysis, due to the fact that the initial conditions that guarantee the reliability of the parametric tests are not satisfied. The techniques presented here can help to cover these necessities, providing the research community with reliable and effective tools for incorporating a statistical analysis into the experimental methodologies. Furthermore, in the KEEL Software Tool [52,53], researchers can find a module for nonparametric statistical analysis, which implements most of the procedures shown in this survey.

Acknowledgements

This work was supported by Project TIN2008-06681-C06-01. J. Derrac holds a research scholarship from the University of Granada.

Appendix. Table for Multiple Comparison Sign test

See Table A.21.

Table A.21
Critical values of minimum r_j for comparison of $m = k - 1$ algorithms against one control in n problems. Source: A.L. Rhyne, R.G.D. Steel, Tables for a treatments versus control multiple comparisons sign test, Technometrics 7 (1965) 293–306.

n	Level of significance (α)	$m = 2$	$m = 3$	$m = 4$	$m = 5$	$m = 6$	$m = 7$	$m = 8$	$m = 9$
5	0.1	0	0	–	–	–	–	–	–
	0.05	–	–	–	–	–	–	–	–
6	0.1	0	0	0	0	0	–	–	–
	0.05	0	0	–	–	–	–	–	–
7	0.1	0	0	0	0	0	0	0	0
	0.05	0	0	0	0	–	–	–	–
8	0.1	1	1	0	0	0	0	0	0
	0.05	0	0	0	0	0	0	0	0
9	0.1	1	1	1	1	0	0	0	0
	0.05	1	0	0	0	0	0	0	0
10	0.1	1	1	1	1	1	1	1	1
	0.05	1	1	1	0	0	0	0	0
11	0.1	2	2	1	1	1	1	1	1
	0.05	1	1	1	1	1	1	0	0
12	0.1	2	2	2	2	1	1	1	1
	0.05	2	1	1	1	1	1	1	1
13	0.1	3	2	2	2	2	2	2	2
	0.05	2	2	2	1	1	1	1	1
14	0.1	3	3	2	2	2	2	2	2
	0.05	2	2	2	2	2	2	1	1
15	0.1	3	3	3	3	3	2	2	2
	0.05	3	3	2	2	2	2	2	2
16	0.1	4	3	3	3	3	3	3	3
	0.05	3	3	3	3	2	2	2	2
17	0.1	4	4	4	3	3	3	3	3
	0.05	4	3	3	3	3	3	2	2
18	0.1	5	4	4	4	4	4	3	3
	0.05	4	4	3	3	3	3	3	3
19	0.1	5	5	4	4	4	4	4	4
	0.05	4	4	4	4	3	3	3	3
20	0.1	5	5	5	5	4	4	4	4
	0.05	5	4	4	4	4	4	3	3
21	0.1	6	5	5	5	5	5	5	5
	0.05	5	5	5	4	4	4	4	4
22	0.1	6	6	6	5	5	5	5	5
	0.05	6	5	5	5	4	4	4	4
23	0.1	7	6	6	6	6	5	5	5
	0.05	6	6	5	5	5	5	5	5
24	0.1	7	7	6	6	6	6	6	6
	0.05	6	6	6	5	5	5	5	5
25	0.1	7	7	7	7	6	6	6	6
	0.05	7	6	6	6	6	6	5	5
30	0.1	10	9	9	9	8	8	8	8
	0.05	9	8	8	8	8	8	7	7
35	0.1	12	11	11	11	10	10	10	10
	0.05	11	10	10	10	10	9	9	9
40	0.1	14	13	13	13	13	12	12	12
	0.05	13	12	12	12	12	11	11	11
45	0.1	16	16	15	15	15	14	14	14
	0.05	15	14	14	14	14	13	13	13
50	0.1	18	18	17	17	17	17	16	16
	0.05	17	17	16	16	16	16	15	15

References

[1] J. Higgins, Introduction to Modern Nonparametric Statistics, Duxbury Press, 2003.

[2] S. García, A. Fernández, J. Luengo, F. Herrera, A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability, Soft Computing 13 (10) (2009) 959–977.

[3] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of nonparametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization, Journal of Heuristics 15 (2009) 617–644.

[4] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

[5] S. García, F. Herrera, An extension on Statistical Comparisons of Classifiers over Multiple Data Sets for all pairwise comparisons, Journal of Machine Learning Research 9 (2008) 2677–2694.

[6] P. Suganthan, N. Hansen, J. Liang, K. Deb, Y. Chen, A. Auger, S. Tiwari, Problem definitions and evaluation criteria for the CEC'2005 special session on real parameter optimization, Nanyang Technological University, Tech. Rep., 2005, Available in http://www.ntu.edu.sg/home/epnsugan/index_files/cec-05/Tech-Report-May-30-05.pdf.

[7] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of IV IEEE International Conference on Neural Networks, Piscataway, New Jersey, 1995, pp. 1942–1948.

[8] A. Auger, N. Hansen, A restart CMA evolution strategy with increasing population size, in: Proceedings of the 2005 IEEE Congress on Evolutionary Computation, 2005, pp. 1769–1776.

[9] L.J. Eshelman, The CHC adaptative search algorithm: How to have safe search when engaging in nontraditional genetic recombination, in: G.J.E. Rawlins (Ed.), Foundations of Genetic Algorithms, Morgan Kaufmann, San Mateo, California., 1991, pp. 265–283.

[10] L.J. Eshelman, J.D. Schaffer, Real-coded genetic algorithms and interval-schemata, in: D. Whitley (Ed.), Foundations of Genetic Algorithms, Morgan Kaufmann, San Mateo, California, 1993, pp. 187–202.

[11] C. Fernandes, A. Rosa, A study of non-random matching and varying population size in genetic algorithm using a royal road function, in: Proceedings of the 2001 Congress on Evolutionary Computation, Piscataway, New Jersey, 2001, pp. 60–66.

[12] H. Mülenbein, D. Schlierkamp-Voosen, Predictive models for the breeding genetic algorithm in continuous parameter optimization, Evolutionary Computation 1 (1993) 25–49.

[13] M. Laguna, R. Martí, Scatter Search. Methodology and Implementation in C, Kluwer Academic Publishers, 2003.

[14] F. Herrera, M. Lozano, D. Molina, Continuous scatter search: An analysis of the integration of some combination methods and improvement strategies, European Journal of Operational Research 169 (2) (2006) 450–476.

[15] K.V. Price, M. Rainer, J.A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, Springer-Verlag, 2005.

[16] A.K. Qin, P.N. Suganthan, Self-adaptive differential evolution algorithm for numerical optimization, in: Proceedings of the 2005 IEEE Congress on Evolutionary Computation, vol. 2, 2005, pp. 1785–1791.

[17] T. Bartz-Beielstein, Experimental Research in Evolutionary Computation: The New Experimentalism, Springer, New York, 2006.

[18] D. Ortiz-Boyer, C. Hervás-Martínez, N. García-Pedrajas, Improving crossover operators for real-coded genetic algorithms using virtual parents, Journal of Heuristics 13 (2007) 265–314.

[19] W. Conover, Practical Nonparametric Statistics, 3rd ed., Wiley, 1998.

[20] M. Hollander, D. Wolfe, Nonparametric Statistical Methods, 2nd ed., Wiley-Interscience, 1999.

[21] R.A. Fisher, Statistical Methods and Scientific Inference, 2nd ed., Hafner Publishing Co, 1959.

[22] D.J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, 4th ed., Chapman & Hall/CRC, 2006.

[23] C. García-Martínez, M. Lozano, Evaluating a local genetic algorithm as context-independent local search operator for metaheuristics, Soft Computing 14 (10) (2010) 1117–1139.

[24] J.D. Gibbons, S. Chakraborti, Nonparametric Statistical Inference, 5th ed., Chapman & Hall, 2010.

[25] J.H. Zar, Biostatistical Analysis, Prentice Hall, 2009.

[26] A. Rhyne, R. Steel, Tables for a treatments versus control multiple comparisons sign test, Technometrics 7 (1965) 293–306.

[27] R. Steel, A multiple comparison sign test: treatments versus control, Journal of American Statistical Association 54 (1959) 767–775.

[28] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, Journal of the American Statistical Association 32 (1937) 674–701.

[29] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Annals of Mathematical Statistics 11 (1940) 86–92.

[30] R. Iman, J. Davenport, Approximations of the critical region of the friedman statistic, Communications in Statistics 9 (1980) 571–595.

[31] J. Hodges, E. Lehmann, Ranks methods for combination of independent experiments in analysis of variance, Annals of Mathematical Statistics 33 (1962) 482–497.

[32] W. Kruskal, W. Wallis, Use of ranks in one-criterion variance analysis, Journal of the American Statistical Association 47 (1952) 583–621.

[33] D. Quade, Using weighted rankings in the analysis of complete blocks with additive block effects, Journal of the American Statistical Association 74 (1979) 680–683.

[34] M. Abramowitz, Handbook of Mathematical Functions, With Formulas, Graphs, and Mathematical Tables, Dover Publications, 1974.

[35] W. Daniel, Applied Nonparametric Statistics, 2nd ed., Duxbury Thomson Learning, 2000.

[36] S.Y.P.H. Westfall, Resampling-based Multiple Testing: Examples and Methods for p-Value Adjustment, John Wiley and Sons, 2004.

[37] O. Dunn, Multiple comparisons among means, Journal of the American Statistical Association 56 (1961) 52–64.

[38] S. Holm, A simple sequentially rejective multiple test procedure, Scandinavian Journal of Statistics 6 (1979) 65–70.

[39] M.C.B.S. Holland, An improved sequentially rejective Bonferroni test procedure, Biometrics 43 (1987) 417–423.

[40] H. Finner, On a monotonicity problem in step-down multiple test procedures, Journal of the American Statistical Association 88 (1993) 920–923.

[41] Y. Hochberg, A sharper Bonferroni procedure for multiple tests of significance, Biometrika 75 (1988) 800–803.

[42] G. Hommel, A stagewise rejective multiple test procedure based on a modified Bonferroni test, Biometrika 75 (1988) 383–386.

[43] D. Rom, A sequentially rejective test procedure based on a modified Bonferroni inequality, Biometrika 77 (1990) 663–665.

[44] J. Li, A two-step rejection procedure for testing multiple hypotheses, Journal of Statistical Planning and Inference 138 (2008) 1521–1527.

[45] K. Doksum, Robust procedures for some linear models with one observation per cell, Annals of Mathematical Statistics 38 (1967) 878–883.

[46] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, Information Sciences 180 (2010) 2044–2064.

[47] P.B. Nemenyi, Distribution-free Multiple comparisons, Master's thesis, Princeton University, 1963.

[48] J. Shaffer, Modified sequentially rejective multiple test procedures, Journal of American Statistical Association 81 (1986) 826–831.

[49] G. Bergmann, G. Hommel, Improvements of general multiple test procedures for redundant systems of hypotheses, in: P. Bauer, G. Hommel, E. Sonnemann (Eds.), Multiple Hypotheses Testing, Springer, 1988, pp. 100–115.

[50] G. Hommel, G. Bernhard, A rapid algorithm and a computer program for multiple test procedures using procedures using logical structures of hypotheses, Computer Methods and Programs in Biomedicine 43 (1994) 213–216.

[51] S. Wright, Adjusted p-values for simultaneous inference, Biometrics 48 (1992) 1005–1013.

[52] J. Alcalá-Fdez, L. Sánchez, S. García, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernández, F. Herrera, KEEL: a software tool to assess evolutionary algorithms for data mining problems, Soft Computing 13 (3) (2008) 307–318.

[53] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework, Journal of Multiple-Valued Logic and Soft Computing 17 (2–3) (2011) 255–287.