Applied Soft Computing 54 (2017) 23–45

Contents lists available at ScienceDirect

Applied Soft Computing

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

On the influence of the number of algorithms, problems, and independent runs in the comparison of evolutionary algorithms

Niki Veček*, Matej Črepinšek, Marjan Mernik

University of Maribor, Faculty of Electrical Engineering and Computer Science, Smetanova 17, 2000 Maribor, Slovenia

ARTICLE INFO

Article history:
Received 10 December 2015
Received in revised form
26 November 2016
Accepted 9 January 2017
Available online 12 January 2017

Keywords:
Multiple comparison
Friedman test
Nemenyi test
CRS4EAs

ABSTRACT

When conducting a comparison between multiple algorithms on multiple optimisation problems it is expected that the number of algorithms, problems and even the number of independent runs will affect the final conclusions. Our question in this research was to what extent do these three factors affect the conclusions of standard Null Hypothesis Significance Testing (NHST) and the conclusions of our novel method for comparison and ranking the Chess Rating System for Evolutionary Algorithms (CRS4EAs). An extensive experiment was conducted and the results were gathered and saved of $k = 16$ algorithms on $N = 40$ optimisation problems over $n = 100$ runs. These results were then analysed in a way that shows how these three values affect the final results, how they affect ranking and which values provide unreliable results. The influence of the number of algorithms was examined for values $k = \{4, 8, 12, 16\}$, number of problems for values $N = \{5, 10, 20, 40\}$, and number of independent runs for values $n = \{10, 30, 50, 100\}$. We were also interested in the comparison between both methods – NHST's Friedman test with post-hoc Nemenyi test and CRS4EAs – to see if one of them has advantages over the other. Whilst the conclusions after analysing the values of k were pretty similar, this research showed that the wrong value of N can give unreliable results when analysing with the Friedman test. The Friedman test does not detect any or detects only a small number of significant differences for small values of N and the CRS4EAs does not have a problem with that. We have also shown that CRS4EAs is an appropriate method when only a small number of independent runs n are available.

1. Introduction

The analysis of an experiment within the field of evolutionary computation is usually conducted by using the Null Hypothesis Significance Testing (NHST), which was developed by Fisher [1–3], and Neyman and Pearson [4]. Even though this method is well-established and used widely, it contains some features that researchers often overlook or ignore. Such features are the number of algorithms participating in comparison k, the number of the optimisation problems N for which the performance is analysed, and the number of independent runs n. In [5] they have already pointed out that "some aspects such as the number of algorithms, number of data sets and differences in performance offered by the control method are very influential in the statistical tests studied" and "an appropriate number of algorithms in contrast with an appropriate number of case problems need to be used in order to employ each type of test". Hence, we have expected that statistical tests depend on the values of these three factors, and one of our research questions in this paper was how much do they affect the detected significant differences and rankings of the algorithms?

The analysis made with the Friedman test and post-hoc Nemenyi test was compared with the analysis made with our novel method the Chess Rating System for Evolutionary Algorithms (CRS4EAs) [6,7]. Our main research question was how different number of algorithms k, number of problems N, and number of independent runs n affect the results and found significant differences for both methods – NHST and CRS4EAs. As the statistical tests have already been reviewed in the past and our method is still very young, we also wanted to obtain a better outlook over CRS4EAs. The comparison between both methods showed that whilst k does not have a drastic effect over rankings, it does affect the number of found significant differences. If k is too small, the found significant differences are unreliable, as was already pointed out by [5] (i.e., "when the number of algorithms for comparison is small, this may pose a disadvantage"). On the other hand, the value of N affects the rankings and number of detected significant differences drastically when analysing with the Friedman ranking and the Nemenyi test.

* Corresponding author.
E-mail addresses: niki.vecek@student.um.si (N. Veček), matej.crepinsek@um.si (M. Črepinšek), marjan.mernik@um.si (M. Mernik).

http://dx.doi.org/10.1016/j.asoc.2017.01.011
1568-4946/

In contrast when analysing with CRS4EAs it affects only the rankings, whilst the number of detected significant differences stays almost the same. As already mentioned above, many suggestions on setting the values of all three factors have already been discussed in the past for statistical methods. Hence, the goal of this paper was not to suggest which values are more appropriate and which less, but to examine the influences of different values. On the other hand, analysis with CRS4EAs, which was the focus of our investigation, showed that there are no special limitations when calculating ratings and ranking evolutionary algorithms using this method, as final results become more reliable with more comparisons.

Hence, the main contributions of this paper are: (1) extensive research from which the effects of the number of algorithms, number of problems, and number of independent runs are analysed with two methods for comparison, (2) comparison of both methods, (3) showing that the results of both methods give similar conclusions regarding different k, (4) showing that the value of N affects NHST more than CRS4EAs, and (5) showing that when n is small CRS4EAs provides more reliable conclusions than NHST.

The paper is organised as follows. Section 2 reviews one common method for statistical comparison and our novel method for the comparison and ranking of evolutionary algorithms. Section 3 displays an extensive experiment and is divided into three sections: (i) effect of the number of algorithms, (ii) effect of the number of problems, and (iii) effect of the number of independent runs when comparing and ranking evolutionary algorithms in multiple $k \times k$ comparison. The paper is concluded in Section 4.

2. Background

A method for multiple comparisons has to be applied in order to distinguish between the performances of two or more algorithms. The common method is NHST [8,9,5,4]. In NHST the null hypothesis, which states that there are no differences in the performances of algorithms, is formed using an alternative hypothesis that states otherwise. The null hypothesis is then rejected or retained using the statistical test of significance (with significance level α). There are many different tests of significance and the choice depends on the data characteristics, the number of algorithms, the number of optimisation problems, and the number of total runs. A more common distinction is regarding the distribution of the data – a parametric test should be used for normally distributed data and a non-parametric test when the distribution is unknown. If more than two algorithms are to be compared (a) the control algorithm can be compared to all the other k algorithms ($1 \times k$ comparison) or (b) all the algorithms can be compared pairwisely ($k \times k$ comparison). This paper focuses on the latter. A more common test for such a comparison is the Friedman test [10,11] or its less conservative variant the Iman and Davenport test [12]. In the Friedman test we rank the mean values of the algorithms (k of them) grouped by problems (N of them) and calculate the average ranking. For further analysis a post-hoc statistical test is required to detect whether there are any significant differences between average rankings. The more common post-hoc test for $k \times k$ comparison is the Nemenyi test [13]. The critical difference CD is calculated using the formula in Eq. (1), where q_α is the critical value based on the Studentised range statistic divided by $\sqrt{2}$ (Table 1) and can be found in various statistical tables (e.g., [14]). The values of CD regarding the different number of algorithms (k) and problems (N) can be seen in Fig. 1.

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{1}$$

If the average rankings differ by more than CD, the null hypothesis can be rejected and it can be concluded that these algorithms are significantly different in their performances.

The detected significant differences in the Nemenyi test depend on the critical difference CD. If CD is smaller, more significant differences will be detected than if CD is larger. Observing the formula in Eq. (1) and Fig. 1, it can be concluded that the value of CD depends on the number of algorithms k, the number of problems N, and the critical value q_α. Garcia et al. [15,16,5] concluded that there must be a right balance between k and N, as these variables affect the detected significant differences. For example, N should not be smaller than $2k$ or bigger than $8k$, or for certain tests, such as Wilcoxon's test [17], N should not be larger than 30. These suggestions are coloured in black colour (in contrast to the light grey colour that denotes areas that are undesirable) in Fig. 1. The overall sample size – which is in multiple comparison a product of the number of problems N and the number of independent runs n – should be as large as possible [15] in order to obtain more reliable mean values. However, because it seems that researchers are selecting the number of algorithms to be compared and the number of problems to solve imprecisely (many times based on the previous experiments), the goal of this paper was to test how k, N, and n really affect the results of experiments/research. Is it possible to select them in a way that the results are in our favour? And more importantly, how selections of k, N, and n affect our novel method for the comparison and ranking of evolutionary algorithms – CRS4EAs.

CRS4EAs [6] is a method based on the Glicko-2 chess rating system [18,19] that was introduced as an improved version of the Glicko rating system [20]. It is based on the Bradley–Terry [21,22] probability model for comparison. Chess is a strategic game between two players that can have three different outcomes: the first player can win and the second loses, the first player can lose and the second wins, and players can play a draw. Chess games are usually organised in a tournament between different players who play pairwisely, which was also an idea that we have incorporated for comparisons between different evolutionary algorithms. In CRS4EAs each algorithm plays the role of a chess player, each comparison between the obtained outcomes of two algorithms is treated as one game, and the pairwise comparison of algorithms is treated as a tournament. The result of a game depends on the solutions two algorithms find of an optimisation problem. If the solution of the first algorithm is closer to the optimum than the solution of the second algorithm, then the first algorithm wins. If the solution of the second algorithm is closer to the optimum than the solution of the first algorithm, then the second algorithms wins. But if the difference between these two solutions is smaller than a predefined ϵ, these two algorithms play a draw. Just like in chess, each algorithm gets information about its power or how well it is performing in comparison to other algorithms [23]. In the Glicko-2 system this power is determined by rating R, rating deviation RD, rating volatility σ, and rating interval RI which is formed from the rating and rating deviation. The values of these variables are calculated with the formulae for the Glicko-2 system regarding the

Table 1
Critical values q_α for Nemenyi test.

k	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$q_{0.10}$	2.052	2.291	2.459	2.589	2.693	2.780	2.855	2.920	2.978	3.030	3.077	3.120	3.159	3.195
$q_{0.05}$	2.343	2.569	2.728	2.850	2.949	3.031	3.102	3.164	3.219	3.268	3.313	3.354	3.391	3.426
$q_{0.01}$	2.913	3.113	3.254	3.364	3.452	3.526	3.591	3.647	3.696	3.741	3.782	3.818	3.852	3.884