RESEARCH ARTICLE OPEN ACCESS

Statistical Methodology for Quality Comparison between Algorithms

Suárez Díaz Ronald[1], Maldonado Ascanio Erik[2], Escorcia Caballero Juan[3]
[1](Professor and Researcher, Department of Industrial Engineering, Universidad Simón Bolívar, Colombia)
[2] (Professor and Researcher, Department of Industrial Engineering, Universidad Autónoma del Caribe, Colombia)
[3] (Professor and Researcher, Department of Industrial Engineering, Universidad Autónoma del Caribe, Colombia)

ABSTRACT
Performance on the quality of the responses obtained by two algorithms, it is not something to be taken lightly. Most of the tests used, are based on the calculation of average, however, this statistic can be affected by extreme results. In addition, a mean difference may be statistically insignificant in practical terms can be decisive. In this paper we propose a statistical test that solved the mentioned problems. The proposed methodology in this research also determines the sample size and the probability that there is significant difference when really there is not (Type II error), factors that generally are not considered in traditional tests.***Keywords*-**Comparison of Algorithms, Number of instances, Optimization, Statistical test, Type II error.

I. INTRODUCTION

It is common in the literature, when comparing the performance of algorithms is to use a mean test, mainly a paired t-student test, however, this method may present some drawbacks from the statistical viewpoint, since by using a paired t-student test without verifying the normality assumption in the samples can lead to significant biases in the conclusions. Further that in some cases is not taken into account the percentage of times that exceeds another algorithm when solving a specific problem. Theseinconvenientcan be explained by the fact that the paired t-student test resulted in no significant difference between the results obtained by both algorithms for a particular problem, but such a conclusion could be affected by the presence of outliers (for example, one or more very large differences in absolute value). Similarly, the presence of some extremely large differences can give false evidence that an algorithm outperforms another, when in fact this may not be so. In optimization problems, the statistical differences considered not significant may prove to be dangerous, because if we are talking for example, of thousands of dollars, two meanslike 985,000 and 965,000 may be statistically equal, but in monetary terms, 20 million dollars is extremely significant.

II. LITERATURE REVIEW

Generally, when comparing two algorithms, one of the most commonly used statistical measures is the mean, eventually leading to the use of a t-student test (Shilane et al, 2008). However, for the case that compares 3 or more algorithms, it is more convenient to make a modification in the calculation of the test statistical (Kenward & Roger, 1997). Other multiple comparison methods are treated by Steel & Torrie (1980), Hochberg & Tamhane (1987), Shaffer (1995), Sokal & Rohlf (1995), Hsu (1996) and Clever & Scarisbrick, (2001). Besides the use of the average for the comparison of algorithms, other measures of interest are the variance and/or the entropy (Rogers & Hsu, 2001; Piepho, 2004).

In regard to performance measurements used, the most common is the quality measurement algorithm. Authors such as Barr (1995), Eiben (2002), Bartz & Beielstein (2004), Birattari (2005) and Hughes (2006) present a list of the different performance measures used in the literature, however, within the measures, no importance is given to the percentage of times that exceeds another algorithm. Peer et al (2003) conducted a study, which shows several cases in which appropriate statistical methods used to analyze results.

Furthermore, some authors report the use of various methods to compare different types of problems. Dietterich (1997) reviews five statistical test for comparing learning algorithms Brazdil et al (2000) compared ranking methods for the selection of classification algorithms, Bouckaert (2003) employs several calibration test for the selection of learning algorithms and Shilane et al (2008) propose a statistical methodology for genetic algorithms performance comparison. In the methodologies cited above the average ranking, ranking of successive radii rate, among others.

In this paper, we illustrate a statistical methodology for the comparison of results between algorithms, taking into account the value of the differences, although not statistically significant, without being affected by this alleged meeting or

outliers. A procedure similar to the one at, is employed in the design of sampling plans for quality control (Montgomery, 2009), but the focus of this investigation is different, and the literature review conducted, indicates that there has been used for comparison of algorithms, or estimate the quality of a particular algorithm.

III. PROPOSED METHODOLOGY

Consider the following set of symbols required to build the theoretical framework of the proposed methodology. The developed method seeks to solve a minimization problem.

n: Numberofinstancesoftheproblemtosolve
f_A: Objectivefunctionvalue bythealgorithmA
f_B: Objectivefunctionvaluebythealgorithm B
$\gamma_{A/B}$: Relative efficiency of Aover B $= \frac{f_B - f_A}{f_A}$
x: Bernoulli random variable $= \begin{matrix} 1, \text{if} \gamma_{A/B} \geq \gamma_0 \\ 0, \text{inothercase} \end{matrix}$
$X = \text{Total number of successes} = \sum_{i=1}^{n} x_i$
p: $P(x = 1)$
α: Type I error
β: Type II error

Thus, instead of inquiring whether an average algorithm outperforms other, we propose to determine if 100p% (strictly equal, at least, or maximum) of the time, an algorithm outperforms other with a minimum relative efficiency γ_0. Since X is the sum of Bernoulli random variables, turns out to be a binomial random variable, whereby this distribution should be used to verify the statistical validity of the assertion with respect to p. Following we describe each of the three cases that can occur.

Case 1: when trying to prove $p \geq p_0$, The approach of the hypothesis would be the following:

$$H_0: p = p_0$$
$$H_1: p < p_0$$
(1)

At first glance, it seems a test of proportions, leading to use the normal distribution (assuming p fits this distribution) to reach a conclusion. However, the idea is not to rely on this assumption and using the Binomial distribution would be really appropriate. If the null hypothesis is true, this should be reflected in the value of X in the sample. A high p value observed should lead to a large value of X, while a small value of p should be reflected in a small value of X. Thus, if the null hypothesis is true, finding a value greater than or equal to X should be very unlikely. Then, the p value of the test proposed in (1) is calculated as:

$$p_{value} = P(X \leq X_0) = \sum_{i=0}^{X_0} \binom{n}{i} {p_0}^i (1 - p_0)^{n-i} \quad (2)$$

Then, H_0 must be rejected if $p_{value} \leq \alpha$.

Case 2: when trying to prove $p \leq p_0$, for this case, we have:

$$H_0: p = p_0$$
$$H_1: p > p_0$$
(3)

The p value corresponding to this case is:

$$p_{value} = P(X \geq X_0) = \sum_{i=X_0}^{n} \binom{n}{i} {p_0}^i (1 - p_0)^{n-i} \quad (4)$$

The rejection criteria is identical to the previous case.

Case 3: when trying to prove $p = p_0$, for this case, we have:

$$H_0: p = p_0$$
$$H_1: p \neq p_0$$
(5)

For this case, if there is no evidence to support the null hypothesis, we observe or be a very small value of X, or a very large value. Thus, there are two sub-cases:

1) $X \leq n/2$. since, at the beginning, is unknown in what sense is the deviation of x, we have:

$$p_{value} = P(X \leq X_0) = 2 \sum_{i=0}^{X_0} \binom{n}{i} {p_0}^i (1 - p_0)^{n-i} \quad (6)$$

2) $X \geq n/2$. we have:

$$p_{value} = 2P(X \geq X_0) = 2 \sum_{i=X_0}^{n} \binom{n}{i} {p_0}^i (1 - p_0)^{n-i} \quad (7)$$

Example: Suppose two algorithms were used to solve 10 instances of the same minimization problem. The results obtained were as follows:

Table 1. Example for the comparison of algorithms

Instance	1	2	3	4	5
f_A	1875	875	1335	542	852
f_B	1897	924	842	624	892
Instance	6	7	8	9	10
f_A	1854	185	2583	3541	365
f_B	1896	120	2612	3663	369

We want to determine if the algorithm A exceeds at least 80% of the time the algorithm B, with a 95% confidence level. If used a paired t-student test and if we verify the equality of means (a different hypothesis), we would have: $\bar{d} = 6.8; s_d = 144{,}11; T = 0.472$ and $t_{0.025,9} = 1.83$. Since $< t_{0.025,9}$, we cannot reject the null hypothesis, so that the algorithms in average generate the same result. However, in practical problems, what really matters is to see which algorithm gets the best answer in most cases.We can propose the following hypothesis:

$$H_0: p = 0.8$$
$$H_1: p < 0.8$$

From the information, we have $\gamma_0 = 0$, from which it follows$X_0 = 8$. Thus:

$$p_{value} = P(X \leq 8) = \sum_{i=0}^{8} \binom{10}{i} 0.8^i 0.2^{10-i}$$
$$= 0.62 > 0.05$$

Therefore, we cannot reject the null hypothesis, so that with a 95% confidence, it is concluded that the algorithm A exceeds the algorithm B at least 80% of the time.

3.1 Type II error and Number of instances

Generally, researchers arbitrarily choose the number of instances that apply to algorithms to compare, which can carry with it a potentially dangerous type II error. Suppose that the test proposed in (1), the true value of p is at least p'_0, where $p'_0 < p$. Let r be the critical value of the test:

$$P(X \leq r) = \alpha = \sum_{i=0}^{r} \binom{n}{i} {p_0}^i (1-p_0)^{n-i} \quad (8)$$

Since in a hypothesis test, β is the probability of not rejecting the null hypothesis when it is false, we have:

$$\beta = P(X > r / p \geq p'_0) = \sum_{i=r+1}^{n} \binom{n}{i} {p_0}^i (1-p_0)^{n-i} \quad (9)$$

(8) and (9) representa system of two non-linear equations with two unknown variables, r and$n$. Since the system is extremely difficult to solve, two phases will be used to solve it:

Phase 1: Normal approximation
Assuming an initial approximation of the binomial distribution to the normal distribution, we have the following parameterization:

$$Z = \frac{x - np}{\sqrt{np(1-p)}} \quad (10)$$

Thus, using jointly (8), (9) and (10), we have:

$$Z_\alpha = \frac{r - np_0}{\sqrt{np_0(1-p_0)}} \quad (11)$$
$$Z_{1-\beta} = \frac{r - np'_0}{\sqrt{np'_0(1-p'_0)}} \quad (12)$$

Isolating r from (11) and (12):

$$Z_\alpha \sqrt{np_0(1-p_0)} + np_0 = Z_{1-\beta} \sqrt{np'_0(1-p'_0)} + np'_0 \quad (13)$$

Associating terms, we have:

$$\sqrt{n}\left(Z_\alpha \sqrt{p_0(1-p_0)} - Z_{1-\beta} \sqrt{p'_0(1-p'_0)}\right) = n(p'_0 - p_0)$$

$$\sqrt{n} = \frac{\left(Z_\alpha \sqrt{p_0(1-p_0)} - Z_{1-\beta} \sqrt{p'_0(1-p'_0)}\right)}{(p'_0 - p_0)}$$

$$n' = \frac{\left(Z_\alpha \sqrt{p_0(1-p_0)} - Z_{1-\beta} \sqrt{np'_0(1-p'_0)}\right)^2}{(p'_0 - p_0)^2} \quad (14)$$

Since (14) can calculate an approximation only, we should look for the exact number of instances to run. This is done in phase two.

Phase 2: Calculation for the number of instances
Then, we propose an algorithm to find the appropriate value of the number of instances that ensures desired significance and statistical power in the analysis.
Step 1: make n = n′
Step 2: Calculate a and b such that:

$$a = \{\min x / \sum_{i=0}^{x} \binom{n}{i} {p_0}^i (1-p_0)^{n-i} \leq \alpha\}$$

$$b = \{\min x / \sum_{i=0}^{x} \binom{n}{i} {p'_0}^i (1-p_0)^{n-i} \geq 1-\beta\}$$

$$If\ a = b\ end, else, make\ n = n+1\ and\ repeat\ step$$

Phase 1 avoids unnecessary scans, as it establishes a minimum limit to the number of instances to be executed.

3.2 Performance Measure of an Algorithm

Besides being useful for proper comparison of algorithms in optimization problems, the test can be modified to measure how well an algorithm performs compared to the optimal solution of the problem (if the latter is known). Under the assumption that this is a minimization problem, let f_{OPT} be the minimum of a particular instance. Thus, we define the following:

$$\gamma: relative\ deviation = \frac{f_A - f_{OPT}}{f_{OPT}}$$

Redefining x:

$$x = \begin{matrix} 1, if\ \gamma \leq \gamma_0 \\ 0, in\ other\ case \end{matrix}$$

$$X = \sum_{i=0}^{n} x_i$$

Thus, the methodology allows us to study, with some reliability, the percentage of times that the algorithm is at a maximum distance of the optimal solution of a problem.
Must be clarified an extremely important point, in regard to the scope of the conclusion of the test. When working with instances of very different sizes, it is advisable to group them by type (small, medium and large) and make an Anova Test first to determine whether the size of the instance is a factor that affects the efficiency of the response obtained by the algorithm. If so, the methodology proposed here should be applied only to relatively homogeneous size instances, so that the conclusions are only valid for instances of that range. If on the contrary, the variance

analysis revealed that the quality of the algorithm is independent of the size of the instance, then, the test results can be generalized.

IV. CONCLUSION

In this paper, we present a statistical methodology for the comparison of algorithms used to solve optimization problems. The methodology does not depend on compliance with any statistical assumption for the accuracy of their results, and allows us to establish with any degree of reliability, the percentage of times that an algorithm outperforms another. This statistic test, under minor modifications, can be extended to measure the performance of an algorithm compared to the optimal response to the problem. The scope of the conclusions depends on whether the quality of the response of the algorithm is affected or not by the size of instances resolved.

REFERENCES

[1] Barr, R., Golden, R., Kelly, J., Rescende M., Stewart, W. Designing and Reporting on Computational Experiments with Heuristic Methods, Journal of Heuristics, 1995.

[2] Bartz, T., Beielstein, T. Design and Analysis of Optimization Algorithms Using Computational Statistics. AppliedNumericalAnalysis& Computational Mathematics, 2004.

[3] Birattari, M., Dorigo, M. How to assess and report the performance of a stochastic algorithm on a benchmark problem: Mean or best result on a number of runs? IRIDIA, Universite Libre de Bruxelles, 2005.

[4] Bouckaert, R. Choosing between two learning algorithms based of calibrate test. Twentieth International Conferenceon Machine Learning. Washington DC, 2003.

[5] Brazdil, P, Soarez, C. A Comparison of Ranking Methods for Classification Algorithm Selection. Faculty of Economics, University of Porto, Portugal.

[6] Clever, A.G., Scarisbrick, D.H., 2001. Practical Statistics and Experimental Design for Plant and Crop Science. Wiley, New York.

[7] Dietterich, T. Approximate Statistical Test for Comparing Supervised Classification Learning Algorithms. Corvallis, OregonStateUniversity, 1993.

[8] Eiben, A. E., Jelasity, M. A critical note on experimental research methodology in EC. Proceedings of the 2002 Congress on Evolutionary Computation (CEC '02), 2002.

[9] Hochberg,Y., Tamhane, A.C., 1987. Multiple Comparison Procedures.Wiley, New York.

[10] Hsu, J.C., 1996. Multiple Comparisons: Theory and Methods. Chapman& Hall, London.

[11] Hughes, E. J. Assessing Robustness of Optimization Performance for Problems with Expensive Evaluation Functions. IEEE CongressonEvolutionaryComputation (CEC 2006), 2006.

[12] Kenward, M.G., Roger, J.H., 1997. Small sample inference for fixed effects from restricted maximum likelihood. Biometrics 53 (3), 983–997.

[13] Montgomey, D., 2009. Statistical Quality Control. Wiley, Arizona.

[14] Peer, E. S., Engelbrecht, A. P., Van den Bergh, F. CIRG@UP OptiBench: A statistically sound framework for benchmarking optimisation algorithms. CongressonEvolutionaryComputation, 2003.

[15] Piepho, H.-P., 2004. An algorithm for a letter-based representation of all-pairwise comparisons. J. Comput. Graph. Statist. 13, 456–466.

[16] Rogers, J.A., Hsu, J.C., 2001. Multiple Comparisons of Biodiversity. Biometrical J. 43 (5), 617–625.

[17] Shaffer, J.P., 1995. Multiple hypothesis testing. Ann. Rev. Psych. 46, 561–584.

[18] Shilane, D., Martikainen, J., Dudoit, S., Ovaska, S. A General Framework for Statistical Performance Comparison of Evolutionary Computation Algorithms. InformationSciences 178 (2008), 2870-2879.

[19] Sokal, R.R., Rohlf, F.J., 1995. Biometry. Freeman, New York.

[20] Steel, R.G.D., Torrie, J.H., 1980. Principles and Procedures of Statistics. McGraw-Hill, New York.