Is "Best-So-Far" a Good Algorithmic Performance Metric?

Nathaniel P. Troutman
Southern Nazarene University
Bethany, Oklahoma, USA
ntroutman@acm.org

Brent E. Eskridge
Southern Nazarene University
Bethany, Oklahoma, USA
beskridge@snu.edu

Dean F. Hougen
School of Computer Science
University of Oklahoma
Norman, Oklahoma, USA
hougen@ou.edu

ABSTRACT

In evolutionary computation, experimental results are commonly analyzed using an algorithmic performance metric called BEST-SO-FAR. While BEST-SO-FAR can be a useful metric, its use is particularly susceptible to three pitfalls: a failure to establish a baseline for comparison, a failure to perform significance testing, and an insufficient sample size. The nature of BEST-SO-FAR means that it is highly susceptible to these pitfalls. If these pitfalls are not avoided, the use of the BEST-SO-FAR metric can lead to confusion at best and misleading results at worst. We detail how the use of multiple experimental runs, random search as a baseline, and significance testing can help researchers avoid these common pitfalls. Furthermore, we demonstrate how BEST-SO-FAR can be an effective algorithmic performance metric if these guidelines are followed.

Categories and Subject Descriptors: I.2.8 Artificial Intelligence: Problem Solving, Control Methods, and Search

General Terms: Experimentation, Algorithms, Performance

Keywords: Empirical study, Genetic algorithms, Machine learning, Performance analysis, Working principles of evolutionary computing

1. INTRODUCTION

Performance graphs, such as the one in Figure 1, are often used to illustrate the effectiveness of evolutionary computation methods, such as genetic algorithms. However, graphing BEST-SO-FAR results in this way can be highly deceptive. Note that Figure 1 shows what appears to be impressive performance for an algorithm. Surprisingly, it is nothing more than the BEST-SO-FAR results for random search. This work focuses on three common pitfalls of some algorithmic performance metrics: failing to establish a baseline for comparison (see Section 2), failing to perform significance testing (see Section 3), and insufficient sample size (see Section 4).

In the literature, raw fitness values for individuals are generally not shown. Instead, results commonly are presented using the BEST-SO-FAR algorithmic performance metric [**?**, 3, 4, 7], which is a metric for comparing the performance of an algorithm, not of a single individual. The analysis of results, comparisons of methods and discussions on an algorithm's traction on a problem, frequently refer to BEST-SO-FAR fitness results. As with other algorithmic performance metrics, misused results from the BEST-SO-FAR metric can be misleading and result in incorrect conclusions.

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-130-9/08/07.

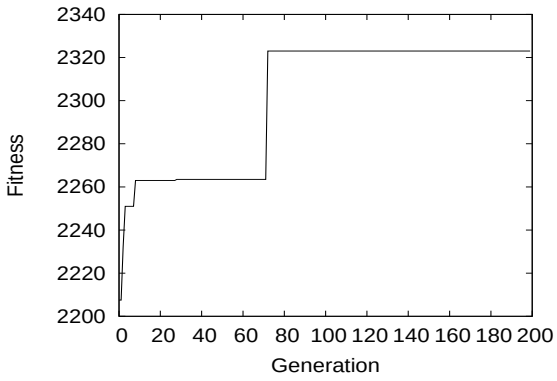

Figure 1: A regular best-so-far curve from an arbitrary problem

2. NO BASELINE COMPARISON

The term *traction* is commonly used when analyzing experimental results. However, what is traction? How is it defined? A plausible definition of traction is finding better and better solutions as learning progresses. Hence, traction on an arbitrary problem using search method X is defined to be: $\text{BSF}_X(g+1) \geq \text{BSF}_X(g)$ where $\text{BSF}_X(g)$ is the BEST-SO-FAR fitness at generation g. Such a simple definition of traction on a problem is of questionable usefulness as BEST-SO-FAR is always monotonically increasing. The particular choice of search method is irrelevant as even random search shows traction on *every* problem.

Instead we propose that traction be defined as an algorithm having a BEST-SO-FAR fitness that outperforms random search. Traction using search method X is defined to be: $\text{BSF}_X(g) \geq \text{BSF}_{random}(g)$ where $\text{BSF}_X(g)$ is the BEST-SO-FAR fitness at generation g.

Traction is now a comparative term, not an absolute one. A particular method, such as a genetic algorithm, would be said to have traction on a problem relative to another method, like random search. This definition of traction requires another method for comparison, hence the need for a baseline method to compare to. Random search is a logical choice since it is the simplest search method and is trivially implemented. Also, any decent search method should outperform random. This new definition of traction provides us with a much stronger and less ambiguous term.

3. NO SIGNIFICANCE TESTING

This new definition of traction has introduced an undefined term. Traction on a problem is said to outperform random, but what does it mean to "outperform" another method? If the results are drawn by a simple visual comparison of plots or of final BEST-SO-FAR fitness values, the conclusion may be unsound and statistically false. A difference between the final BEST-SO-FAR values does not guarantee a difference in the means of the underlying distributions. Performing statistical confidence tests on experimental data will improve the quality of the conclusions. The simplest statistical significance test to use is the Student's t-Test at a minimum of a 95% confidence level to test the final BEST-SO-FAR fitness of all runs [5]. A more sophisticated and detailed approach using a randomized ANOVA is proposed by Piater et al. [9].

4. INSUFFICIENT SAMPLE SIZE

The solution spaces of most problems for which genetic algorithms are applied are too large to practically perform an exhaustive search, thus genetic algorithms search a relatively small number of solutions. The final fitness of a population is dependent upon the individuals in the initial population and the random choices made throughout the run. Thus, a very poor initial population can hamstring the ability of a genetic algorithm to find good solutions [2, 6, 8].

The independence between runs makes it difficult to make a good estimation of the expected performance of on a given problem since there is the potential for a large variance between runs. Error in estimation can come from the samples being clustered together at one extreme of the range of fitness values. This clustering can have a very small variance in the BEST-SO-FAR fitness values. However, this does not actually indicate that the observed median is close to the actual median as the sample consists of outliers. Another source of error in estimation comes from the samples possibly covering a wide range of values. This results in a very large variance, meaning the actual difference between the observed median and actual median is irrelevant as there can be no confidence in the observed median.

Small sample sizes and the stochastic nature of genetic algorithms makes them susceptible to bad statistics. Since a majority of the analysis performed on the experimental results in evolutionary computation is of a statistical nature, it is important that the sample size be large enough for the statistics to be meaningful. It is recommended that multiple runs, at least 30, be done so as to have a sufficiently large sample size and ensure statistical significance [1, 7]. With a large number of runs supporting the results, the conclusions drawn from the results are much better supported.

5. CONCLUSIONS

Many papers in the evolutionary computation literature use the BEST-SO-FAR algorithmic performance metric to analyze their results. It produces pleasing plots by removing the noise that is frequently seen in other metrics such as CURRENT-POPULATION-MEAN or CURRENT-POPULATION-BEST. Intuitively, BEST-SO-FAR is a good algorithmic performance metric. However, there are serious problems if it is used incorrectly. Sense it is monotonically increasing and never shows a degradation in algorithm performance, BEST-SO-FAR can lead to a false sense of success of in regards to an algorithm's performance.

There are three common pitfalls researchers can run into when presenting their experimental results using BEST-SO-FAR. These pitfalls are: no baseline comparison, lack of significance testing, and insufficient sample size. BEST-SO-FAR is particularly susceptible these pitfalls because it is monotonically increasing. The use of a baseline for comparison makes BEST-SO-FAR a meaningful metric for comparisons. Traction on the problem is thus defined as outperforming the baseline. Random search is suggested as a baseline since any viable search method must be able to outperform it. Comparisons made by simply observing the difference of two fitness values are not always accurate since the numbers being compared are from multiple independent runs. Therefore, significance testing, such as the Student's t-test, is important as it indicates whether or not the observed difference is significant. When comparisons between methods are made, a single run is insufficient as there can be no statistical confidence in any conclusions drawn from a single data point. Hence, a large number of runs, at least 30, is needed for statistical confidence in any conclusions. Despite the potential pitfalls of using BEST-SO-FAR, it is still a useful algorithmic performance metric when used cautiously, and is best utilized when comparing two different methods and not as a stand-alone metric.

6. REFERENCES

[1] P. R. Cohen. *Empirical Methods for Artificial Intelligence*. MIT Press, 1995.

[2] P. A. Diaz-Gomez and D. F. Hougen. Initial population for genetic algorithms: A metric approach. In *Proceedings of the 2007 International Conference on Genetic and Evolutionary Methods (GEM'07)*, 2007.

[3] J. Koza. Survey of genetic algorithms and genetic programming. In *Wescon: Microelectronics, Communications Technology, Producing Quality Products, Mobile and Portable Power, Emerging Technologies*. IEEE, New York, NY, 1995.

[4] J. R. Koza and D. Andre. Evolution of iteration in genetic programming. In L. J. Fogel, P. J. Angeline, and T. Baeck, editors, *Evolutionary Programming V: Proceedings of the Fifth Annual Conference on Evolutionary Programming*. MIT Press, 1996.

[5] J. H. S. Lisa Lavoie Harlow, Stanley A. Mulaik. *What If There Were No Significance Tests?* Lawrence Erlbaum Associates, 1997.

[6] F. G. Lobo and C. F. Lima. A review of adaptive population sizing schemes in genetic algorithms. In *Proceedings of the 2005 workshops on Genetic and evolutionary computation*, pages 228–234. ACM, 2005.

[7] S. Luke and L. Panait. Is the perfect the enemy of the good? In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 820–828. Morgan Kaufmann, 2002.

[8] K. P. A. Maaranen, Heikki; Miettinen. On initial populations of a genetic algorithm for continuous optimization problems. In *Journal of Global Optimization*, volume 37, pages 405–436. Springer, March 2007.

[9] J. H. Piater, P. R. Cohen, X. Zhang, and M. Atighetchi. A randomized ANOVA procedure for comparing performance curves. In *Proceedings of the 15th International Conference on Machine Learning*, pages 430–438. Morgan Kaufmann, 1998.