DOI: 10.58895/ksp/1000174544-9 erschienen in:
Proceedings - 34. Workshop Computational Intelligence: Berlin, 21.-22. November 2024
DOI: 10.58895/ksp/1000174544 | https://www.ksp.kit.edu/site/books/e/10.58895/ksp/1000174544/

Investigating Reproducibility of Ultra-High Performance Concrete with Consistent Mechanical Properties: A Modeling Pipeline for Complex Manufacturing

Farzad Rezazadeh P.[1], Amin Abrishambaf[2], Axel Dürrbaum[1], Gregor Zimmermann[2], Andreas Kroll[1]

[1]Department of Measurement and Control, University of Kassel
Mönchebergstr. 7, 34125 Kassel, Germany
E-Mail: farzad.rezazadeh, axel.duerrbaum, akroll@mrt.uni-kassel.de

[2]German Technologies & Engineering Concepts - G.tecz Engineering GmbH
Eichwaldstr. 38, 34125 Kassel, Germany
E-Mail: abrishambaf, zimmermann@gtecz.com

1 Introduction

Ultra-High Performance Concrete (UHPC) is distinguished by its exceptional mechanical properties and durability, offering significant advantages over conventional concrete [1]. However, achieving consistent UHPC quality remains a challenge, even when following the same formulation [2]. This inconsistency is largely due to the production process's sensitivity to variations in raw material properties, environmental conditions, and operational factors (see Figure 1). Adopting a holistic approach to the entire UHPC manufacturing process introduces high dimensionality and increases the cost of data generation, resulting in sparse datasets. To address these challenges, this study presents an automated modeling pipeline (see Figure 2) specifically designed to tackle high-dimensional and sparse data issues. The proposed pipeline comprises five layers: data generation based on a three-phase Design of Experiments (DoE), data preprocessing, ensemble-based feature importance determination [2], feature

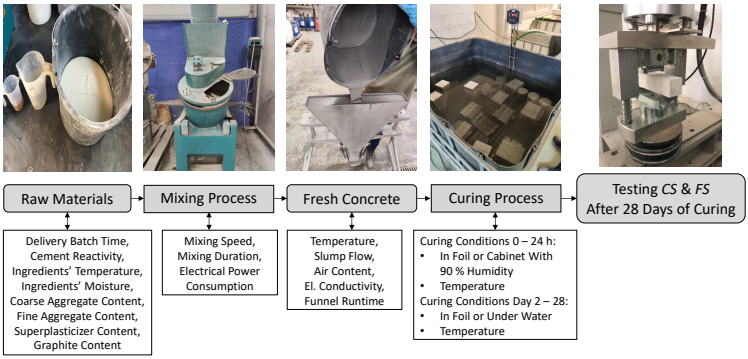


Figure 1: Ultra-High Performance Concrete (UHPC) Production Process [2]. The diagram shows key factors influencing UHPC production and testing. (El: Electrical, CS: Compressive Strength, FS: Flexural Strength)

selection using the Informed Non-Dominated Sorting Genetic Algorithm II (I-NSGA-II), and a modeling layer. The primary objective is to analyze the consistency of key mechanical properties, specifically compressive and flexural strength after 28 days of curing.

2 Modeling Pipeline for the Ultra-High Performance Concrete Production Process

2.1 Data Generation

Due to resource limitations restricting the number of experiments to 100 in this study, a feature pool was developed through a literature review and expert input, ensuring a comprehensive representation of the entire UHPC manufacturing process. To further manage the dimensionality of the input space, feature engineering was employed as an initial strategy to reduce the number of features while preserving critical information.

An effective DoE is crucial for managing complex input spaces. To address this, a three-phase DoE was developed: Screening, Optimal, and Complementary Phases. The L-50 Taguchi Orthogonal Array [3] was used in the Screening

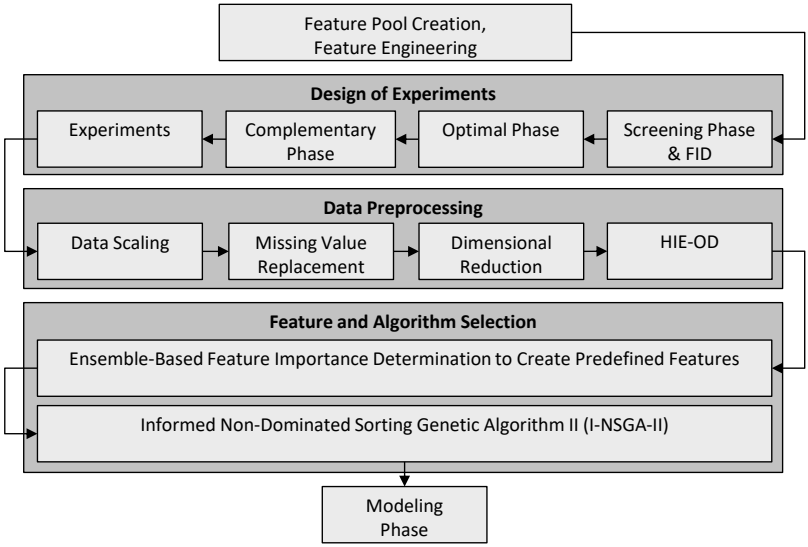


Figure 2: Automated Modeling Pipeline for Ultra-High Performance Concrete Manufacturing: An End-to-End Framework Covering Data Generation to Model Building. (FID: Feature Importance Determination, HIE-OD: Human-in-the-Loop Informed Ensemble-based Outlier Detection)

Phase due to its ability to handle high-dimensional spaces with few observations and ensure uniform coverage of the input space, resulting in 50 data points. This phase led to the removal of three non-essential features and the identification of curing conditions as the most critical feature.

In the Optimal Phase, the goal was to strategically position new data points relative to those established in the Screening Phase. To achieve this, 50 additional experiments were conducted using S-optimality criteria and Latin Hypercube Sampling (LHS) [4], refining the experimental design based on insights gained from the initial phase. Recognizing that curing conditions were the most influential factors, the Complementary Phase was conducted in parallel with the Optimal Phase. In this phase, each experiment designed in the Optimal Phase was replicated under two curing conditions: one based on the Optimal Phase design and the other informed by expert knowledge. This dual-condition approach generated two outputs per experiment, resulting in a total of 150

data points from 100 experiments. This comprehensive dataset ensures robust coverage of the UHPC manufacturing process for subsequent modeling.

2.2 Data Preprocessing

Given the high dimensionality and limited data size, it was crucial to reduce dimensionality while addressing missing values to maintain data integrity. Traditional outlier detection methods were unsuitable for the sparse and uniformly distributed dataset, necessitating the development of a human-in-the-loop informed ensemble-based outlier detection (HIE-OD) method. The preprocessing steps involved using imputation techniques to retain four experiments and remove two highly correlated features as part of dimensionality reduction Additionally, 11 outliers were identified and removed, resulting in a refined dataset with 16 features and 139 observations.

2.3 Feature Importance Determination & Feature Selection

To effectively handle sparse data, as encountered in this study, feature selection is crucial. Traditional grid-search methods are unsuitable for exploring such complex input spaces due to a high risk of converging to local minima, especially in sparse datasets. Therefore, an evolutionary, multi-objective feature selection approach is recommended. These methods not only provide a comprehensive exploration of complex input spaces but also act as a form of regularization. However, a significant challenge with these methods is the instability of feature selection in high-dimensional spaces with limited observations, which can lead to poor predictive performance [5, 6].

To address these issues, we used the Informed Non-Dominated Sorting Genetic Algorithm II (I-NSGA-II) as a multi-objective feature selector. This approach effectively tackles both challenges associated with evolutionary multi-objective feature selection. The I-NSGA-II algorithm incorporates the most important features, as determined by the prior ensemble-based feature importance determination [2], as part of its initial solution. It then searches for additional features that are relevant to predicting the mechanical properties of UHPC,

thereby improving both stability and predictive accuracy. The performance of the proposed I-NSGA-II was compared to the traditional NSGA-II [7] using ten different machine learning algorithms [8].

2.4 UHPC Manufacturing Process Modeling

The UHPC manufacturing process was modeled and evaluated using the leave-one-out cross-validation technique in the final step. This followed the generation of datapoints in the *Design of Experiments* step, the cleaning of data in the *Data Preprocessing* step, and the selection of the most important features and the best algorithm in the *Feature and Algorithm Selection* step.

3 Results and Future Work

The proposed framework was validated on two experimental datasets, proving its effectiveness in addressing challenges related to data dimensionality and limited sample size. Future work will focus on advancing the modeling phase by incorporating more complex and innovative algorithms, as well as testing the framework with additional benchmark functions.

Acknowledgment

This project (HA project no. 1011/21-13) is financed with funds of LOEWE – Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz, Förderlinie 3: KMU-Verbundvorhaben (State Offensive for the Development of Scientific and Economic Excellence).

References

[1] A. Abrishambaf, M. Pimentel, S. Nunes and C. Costa. “Multi-level study on UHPFRC incorporating ECat”. *Construction and Building Materials* 318. p. 125976. 2022.

[2] F. Rezazadeh P., A. Dürrbaum, G. Zimmermann and A. Kroll. “Leveraging ensemble structures to elucidate the impact of factors that influence the quality of ultra–high performance concrete”. In: *IEEE Symposium Series on Computational Intelligence (SSCI)* Mexico City. pp. 180–187. 2023.

[3] G. Taguchi. “System of experimental design: engineering methods to optimize quality and minimize costs”. *UNIPUB/Kraus International Publications, American Supplier Institute* 1987.

[4] M. Stein. “Large sample properties of simulations using Latin hypercube sampling”. *Technometrics* 29.2. pp. 143–151. 1987.

[5] M. Amoozegar and B. Minaei-Bidgoli. “Optimizing multi-objective PSO based feature selection method using a feature elitism mechanism”. *Expert Systems with Applications* 113. pp. 499–514. 2018.

[6] R. Jiao, B.H. Nguyen, B. Xue and M. Zhang. “A survey on evolutionary multiobjective feature selection in classification: approaches, applications, and challenges”. *IEEE Transactions on Evolutionary Computation* p. 1. 2023.

[7] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. *IEEE Transactions on Evolutionary Computation* 6.2. pp. 182–197. 2002.

[8] F. Rezazadeh and A. Kroll. “Predicting the compressive strength of concrete up to 28 days-ahead: Comparison of 16 machine learning algorithms on benchmark datasets”. In: *Proceedings - 32. Workshop Computational Intelligence* Berlin. pp. 53–75. 2022.

A Novel Ranking Scheme for the Performance Analysis of Stochastic Optimization Algorithms using the Principles of Severity

Sowmya Chandrasekaran, Thomas Bartz-Beielstein

Institute for Data Science, Engineering, and Analytics, TH Köln, Steinmüllerallee 1, 51643 Gummersbach, Germany
E-Mail: {sowmya.chandrasekaran,thomas.bartz-beielstein}@th-koeln.de).

Abstract

Stochastic optimization algorithms have been successfully applied in several domains to find optimal solutions. Because of the ever-growing complexity of integrated systems, novel stochastic algorithms are being proposed, which makes the task of the performance analysis of the algorithms extremely important. This paper provides a novel ranking scheme to rank the algorithms over multiple single-objective optimization problems. The results of the algorithms are compared using a robust bootstrapping-based hypothesis testing procedure that is based on the principles of severity. Analogous to the football league scoring scheme, we propose pairwise comparison of algorithms as in league competition. Each algorithm accumulates points and a performance metric of how good or bad it performed against other algorithms analogous to the goal differences metric in the football league scoring system. The goal differences performance metric can be used not only as a tie-breaker but also to obtain a quantitative performance of each algorithm. The key novelty of the proposed ranking scheme is that it takes into account the performance of each algorithm considering the magnitude of the achieved performance improvement along with its practical relevance and does not have any distributional assumptions. To demonstrate the advantages of the proposed ranking scheme, we compare the expected run-time metrics of three hyperparameter optimization (HPO)

procedures, namely, Irace, a mixed-integer parallel efficient global optimization (MIP-EGO), the mixed-integer evolution strategy (MIES), along with (1+1)EA and grid search(GS) on a genetic algorithm framework for Pseudo-Boolean Optimization (PBO) Suite of 25 problems. The proposed ranking scheme is compared to classical hypothesis testing and the analysis of the results shows that the results are comparable and our proposed ranking showcases many additional benefits.

1 Introduction

Numerous new meta-heuristic algorithms are being proposed to solve various complex problems. This makes the analysis of the performances of the algorithms to a relevant set of problems an inevitable task. Generally, the performances of the stochastic optimization algorithms are evaluated based on solution quality or utilized budget [1]. Here, the solution quality measures how close the solution obtained by an algorithm is with respect to the global optimum or the best-known value. This is referred to as the *fixed-budget* measure, where the achievable solution quality for a fixed budget is obtained. In the *fixed-target* perspective, the time required by an algorithm to hit the desired solution quality is measured. The time required can be CPU time or function evaluations. Typically, the CPU time can be dependent on many factors like computing environment, hardware resources, etc. Hence, the Function Evaluations (FE) is considered as an alternative time measure, where the number of times the objective function evaluated is measured. To test the robustness of the algorithm's performances, the algorithms are tested under uncertainty, noise, etc. The scalability measures the ability of the algorithm as the dimension of the problem increases. Be it the fixed-target or fixed-budget measure, due to the stochastic nature of the algorithm, there exists randomness in the performance of algorithms. Executing the same algorithm repeatedly can produce different solutions for the same inputs. Hence, there is a need for rigorous analysis of the performances of stochastic optimization algorithms.

In recent years, descriptive analysis (e.g. mean, median, best, worst and standard deviation) has turned out to be necessary but not sufficient metrics in analysing

the performances of the algorithms. Statistical analysis plays a crucial role in comparing the performances of the algorithms [1]. A commonly used statistical tool over several years is hypothesis testing [15]. In order to compare the performances of algorithms, the null hypothesis can be formulated as *There is no statistically significant performance difference between a pair of algorithms vs* the alternative hypothesis as *There exists a statistically significant performance difference between a pair of algorithms.* Hypothesis testing can be broadly classified into parametric and non-parametric tests. While the former assumes a specific type of probability distribution of the data and makes inferences about the parameters of the distribution, the latter do not make any explicit assumptions about the data or its underlying distributions. The non-parametric tests are used when the assumptions for the safe use of parametric tests are not met. In both procedures, the p-value be the measure for deciding whether to retain or reject the null hypothesis. Since the statistical significance is decided in the form of a yes or no fashion that is based only on the p-value, the hypothesis testing procedure is criticized as black and white thinking [3]. Considering these criticisms, in [20], the American Statistical Association (ASA) explains the scope of p-value, wherein it emphasizes on considering additional appropriate measures along with p-value for a scientific decision. In [7], this issue is addressed using a measure, severity, which is a form of attained power [14]. More precisely, the concept of using severity as a tool for single pairwise comparison is proposed in [7], where both the statistical significance and also the practical relevance of the algorithms performances are measured. Also, the concept of severity is utilized for the analysis of the performances of hyper-parameter tuning in machine and deep learning algorithms [2].

The purpose of this paper is to propose a novel ranking scheme, that ranks the algorithms in a robust statistical fashion based on their performances considering the statistical significance, practical significance and magnitude of the achieved performance improvement. The resulting ranking scheme is analogous to the football league ranking system which has both the *points* scored and goal differences metric. Considering Multiple Algorithm Multiple Problem design (MAMP), each pairwise comparison of algorithms is treated similarly to a football game between two teams in a football league competition. The outcome of each game can be a *win*, a *draw*, or a *loss* for each algorithm on each problem

based on which the *points* and *goal difference* are obtained. To the best of our knowledge, this may be the first statistical ranking scheme, which takes into account the win or loss of an algorithm along with the magnitude of the corresponding win or loss, i.e, in terms of the positive or negative goal differences. At the end of the league competitions, the algorithms are ranked based on the *points* and *goal difference*(GD).

In [7], the severity for various values of discrepancies is evaluated. The choice of evaluating varied discrepancies can be a little time-consuming. To overcome this issue, in this proposed paper, we have introduced a simple yet very effective concept, where the user needs to choose the desired severity, $S \in [0,1]$ (same as desired power). Note, this is the same as the desired power and not an explicit additional parameter. Then, the supported discrepancy for the chosen severity is obtained as an outcome. Based on outcome each algorithm attains two scoring as points and goal differences.

The paper is organized as follows: Section 2 explains the existing ranking frameworks to evaluate the performances of the optimization algorithms. Section 3.1 explains the concept of severity. Section 3.2 summarizes the proposed football based ranking scheme. In Section 4, a specific set of hyperparameter optimization techniques are evaluated for a family of genetic algorithms and are tested using the PBO Suite of 25 problems. Section 5 concludes with a summary and outlook.
Reproducibility: The source code is made available at `https://github.com/sowmyachandrasekaran17/algRanking`. The data and results of the experiment are available in [6].

2 Related Works

There exist some ranking schemes in the literature which were proposed to compare the performances of the optimization algorithms as in [5, 9, 10, 16, 18]. In [18], an empirical chess rating system for evolutionary algorithms using Glicko-2 rating is proposed. Here, the evolutionary algorithms are treated as chess players, and a pairwise comparison of two algorithms is considered as one game. Each game outcome can be a *win*, a *draw*, or a *loss*. At the end of

the tournament, each algorithm is represented by rating(R), rating deviation (RD), and rating volatility(σ). Despite not being statistically analyzed, this rating system suffers from other issues. Firstly, the ordering of the games affects the final rating, though it is randomly selected. Furthermore, the overlapping of the confidence intervals might lead to statistical inconsistency as explained in [9]. Finally, the magnitude of the win or loss is not considered in the rating system.

More recently, different variants of deep statistic-based comparison (DSC) tool have been proposed [9, 10]. The advantage of this [10] ranking scheme is that it is based on the whole distribution rather than comparing mean or medians. In MAMP design, multiple pairwise comparisons of algorithms are performed using the non-parametric Kolmogorov-Smirnov or Anderson-Darling test and only the p-values determine the win or loss. Though the practical significance is addressed in DSC, the magnitude of the win or loss is not considered. Also, since the practical significance is directly included in the hypothesis formulation, the approach can be more conservative.

In [5, 16], the statistical comparison of the performances of the evolutionary algorithms is performed using Bayesian inferences. Though the identification of prior probabilities is an issue [11], the ranking of the algorithms is only based on the Bayesian probability of an algorithm being the best performer. The magnitude of the performance differences cannot be obtained.

3 Proposed Ranking Scheme

3.1 Concept of Severity

In order to explain the concept of hypothesis testing and severity, let us assume Normal, Independent, and Identically Distributed (NIID) data[1]. Let us consider algorithm A, say $\mathbf{a} = (a_1, \ldots, a_n)$, representing the function evaluations required by Algorithm A to achieve a specific target solution for n runs. Similarly,

[1] In the proposed ranking scheme, we do not assume the normality of the data. Here it is assumed to simplify the explanation of the concept and without the loss of generality, the concept can be adapted to the cases where the distribution is not known.

algorithm B, say $\mathbf{b} = (b_1, \ldots, b_n)$, represents the function evaluations required by Algorithm B to achieve the same target solution for n runs. We evaluate the performance of the algorithms repeatedly for n runs to handle the randomness in the evaluation metric and to obtain a reliable estimate of the metric.

The hypothesis testing is performed as an upper tail test of the mean differences as it is an optimization minimization problem. The difference vector $\mathbf{x}$ can be defined as $\mathbf{x} = (x_1, \ldots, x_n)$, where $x_i = a_i - b_i, \forall i = \{1, \ldots, n\}$ and $\bar{\mathbf{x}}$ denote the mean of the vector $\mathbf{x}$.
$H_0 : B$ does not achieve less FE than $A \implies \bar{\mathbf{x}} \leq 0 \implies$ Loss for B.
$H_1 : B$ achieves less FE than $A \implies \bar{\mathbf{x}} > 0 \implies$ Win for B.

$$H_0 : \bar{\mathbf{x}} \leq 0; \text{vs. } H_1 : \bar{\mathbf{x}} > 0,$$

$$decision = \begin{cases} \text{not-Reject} H_0, & \text{if } d(\mathbf{X}) \leq u_{1-\alpha}, \\ \text{Reject} H_0, & \text{otherwise}, \end{cases}$$

where $u_{1-\alpha}$ is the upper tail cut-off point of the normal distribution, which cuts the upper-tail probability of α, and the test statistic $d(\mathbf{X})$ can be represented as

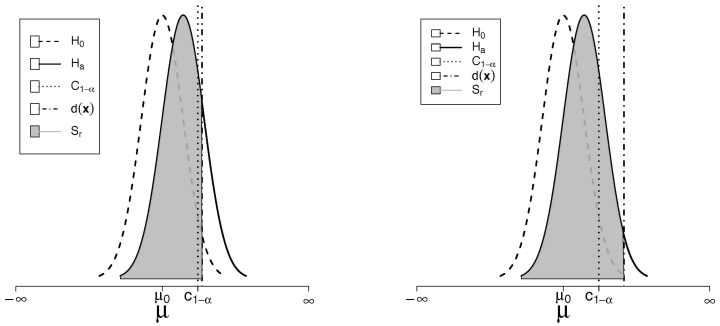
$$d(\mathbf{X}) = \frac{\bar{X} - \mu_0}{\sigma_x},$$

where standard error $\sigma_x = \frac{\sigma}{\sqrt{N}}$ and μ_0 is the hypothesized mean under H_0. If a test statistic is observed beyond the cut-off point, we reject the H_0 at a significance level α. Here, the values for α, β, and hence power $(1 - \beta)$ are pre specified before the experiment is performed.

In the context of ranking the algorithms based on the results of parametric or non-parametric hypothesis testing, based only on the p-value, i.e, if a test statistic is observed beyond the cut-off point, the H_0 is rejected and hence B wins and is declared of achieving less FE than A. Also, B gains a point just based on this decision. This is criticized as black and white thinking.
Severity, a form of attained power [14], is a probability analogous to the p-value under the alternative hypothesis rather than one under the null. In case of win or loss, the magnitude of the performance improvement is measured in terms of severity as S_r and S_{nr} respectively. The loss S_{nr} values increase monotonically

from 0 to 1. The won S_r values decrease monotonically from 1 to 0. The closer the value is to 1, the more reliable the decision made with the hypothesis test. Generally, a severity value of 0.8 is considered reliable support. The differences between α, p-value, power, and severity is provided in Table 1 in [7].

The importance of severity representation of won is shown in Figure 1. In a similar fashion, the severity representation of loss can also be visualized. In Figure 1, only one value for the alternate hypothesis is visualized. However, in practice, we evaluate the severity for specific possible values under the H_1. The different values under the alternative hypothesis for which the compatibility is assessed will henceforth be called discrepancy, δ. It measures how discrepant is the performance improvement when compared to the null improvement. In the context of algorithm ranking, since we consider the FE as the metric, we can evaluate if B won over A with 1000 FE or 10 FE, thereby quantization the magnitude of the victory.



(a) Won Scenario 1 :Severity to Reject H0

(b) Won Scenario 2 :Severity to Reject H0

Figure 1: Illustration of two scenarios of S_r under the alternate hypothesis. In both cases, the actual test statistic $d(\mathbf{x})$, falls outside the $u_{1-\alpha}$, the decision is to reject the null. The S_r is the area under the H_1 that is within the $d(\mathbf{x})$ (shaded area). Though in both cases, the decision is the same, severity sheds light in understanding the actual attained power of the test. In (a), less support for the decision won (shaded area) as $d(\mathbf{x})$ is closer to the cut-off point and in (b), more support for the won (shaded area) as $d(\mathbf{x})$ is way more from the cut-off point.

3.2 Proposed Algorithm

The football league scoring system has been designed to provide a fair and objective method for ranking teams based on their performances. Each football match between two teams, has three possible outcomes: win, loss or draw. Based on the outcome the teams get *points*, 3 for winning, 1 for drawing and 0 for losing. The *points* earned by each team from matches played over a season are cumulatively added and the final rankings for the season is obtained. In addition, the goal difference, which is the difference between the number of goals scored minus the number of goals conceded in matches is calculated. Goal difference serves as a tiebreaker if two or more teams have the same *points*.

Analogous to the football league scoring scheme, we propose a novel ranking scheme for ordering the performances of several stochastic algorithms on a set of well-known bench-marking test functions ($\mathcal{F}$) as Algorithm 1. Let $\mathcal{A} := \{A_i, \forall i \in \{1,\ldots,k\}\}$ denote the set of all algorithms that must be ranked and $\mathcal{F} := \{F_l, \forall l \in \{1,\ldots,m\}\}$ denote the set of all test functions. An experiment is performed where each algorithm $A_i \in \mathcal{A}$ is evaluated on each of the test functions $F_l \in \mathcal{F}$ for n runs. Let $\mathcal{C}$ represent set of all possible pair-wise algorithm comparisons, $\mathcal{Y}$ represent the set of all corresponding solutions and are defined as

$$\mathcal{C} := \big\{(A_i, A_j)_l, \forall i, j \in \{1,\ldots,k\}, i \neq j, \forall l \in \{1,\ldots,m\}\big\}, \tag{1}$$

$$\mathcal{Y} := \big\{(\mathbf{y}_i, \mathbf{y}_j)_l, \forall i, j \in \{1,\ldots,k\}, i \neq j, \forall l \in \{1,\ldots,m\}\big\}, \tag{2}$$

where $\mathbf{y}_i \in \mathbb{R}^n$ denotes the solution of i^{th} algorithm A_i for n runs for a given function in $\mathcal{F}$. The ranking scheme requires each pairwise comparison defined in $\mathcal{C}$ and set $\mathcal{Y}$ should be obtained and this results in a total of $k \times k - 1 \times m$ comparisons. For each $(A_i, A_j)_l \in \mathcal{C}$, bootstrapping-based t-test is performed. Sampling with replacement is done to attain a better estimate of the metric. This procedure also eliminates the dependence of the outcome on the ordering of the optimization runs. For a given value of $i, j \in \{1,\ldots,k\}, l \in \{1,\ldots,m\}, i \neq j$, the following steps are performed.

Algorithm 1.: Proposed Ranking Scheme

for each pair-wise comparison $(A_i, A_j)_l \in \mathcal{C}$ **do**
  Require: $(\mathbf{y}_i, \mathbf{y}_j)_l = ((y_i^1, \ldots, y_i^n), (y_j^1, \ldots, y_j^n))_l$, α, S,n_b, δ_p
  Formulate Hypothesis $H_0 : \bar{\mathbf{x}} \leq 0$ vs. $H_1 : \bar{\mathbf{x}} > 0$
  Evaluate observed sample mean difference $t_{obs} = \overline{\mathbf{y}_i} - \overline{\mathbf{y}_j}$
  Combine $\mathbf{I} = \widehat{\mathbf{y}_i \mathbf{y}_j}$
  repeat
    Draw a bootstrap sample of $2n$ observations with replacement from $\mathbf{I}$
    Let the mean of the first n observation be $\overline{\mathbf{y}_i}^*$
    Let the last n observations be $\overline{\mathbf{y}_j}^*$
    Evaluate $t^{*\mathrm{bs}} = \overline{\mathbf{y}_i}^* - \overline{\mathbf{y}_j}^*$
    Evaluate $t_s^{*\mathrm{bs}} = \overline{\mathbf{y}_i}^* - \overline{\mathbf{y}_j}^* - \delta$
  until n_b times
  Calculate $p \approx \frac{\#(t^{*\mathrm{bs}} \geq t_{obs})}{n_b}$
  Obtain $adjusted - p$ based on BH correction
  if $adjusted - p \leq \alpha$ **then**
    decision: Reject H_0 and $\delta^* = \min_\delta S - \frac{\#(t_s^{*\mathrm{bs}} \leq t_{obs})}{n_b}$
    if $\delta \leq \delta_p$ **then**
      $points = 1$ and $GD = 0$
    else
      $points = 3$ and $GD = \lfloor \frac{\delta^*}{\delta_p} \rfloor$
  if $adjusted - p > \alpha$ **then**
    decision: not-Reject H_0 and $\delta^* = \max_\delta S - \frac{\#(t_s^{*\mathrm{bs}} > t_{obs})}{n_b}$
    $points = 0$ and $GD = \lfloor \frac{\delta^*}{\delta_p} \rfloor$
Obtain cumulative *points* and *GD*
Rank based on *points* and *GD*

1. Merge the results of the algorithms to obtain $\mathbf{I} := \widehat{\mathbf{y}_i \mathbf{y}_j}$ of size $2n$ and compute the observed sample mean difference $t_{obs} := \overline{\mathbf{y}}_i - \overline{\mathbf{y}}_j$.

2. Draw a bootstrap sample of $2n$ observations with replacement from $\mathbf{I}$ and evaluate the bootstrap test statistic $t^{*\mathrm{bs}} = \overline{\mathbf{y}}_i^* - \overline{\mathbf{y}}_j^*$.

3. Repeat the procedure based on the bootstrapping re-sample size n_b, and estimate p-value as the number of times the bootstrap test statistic $t^{*\mathrm{bs}}$ was found to be greater than the t_{obs} in n_b samples. I.e., $p \approx \frac{\#(t^{*\mathrm{bs}} \geq t_{obs})}{n_b}$

4. Adjust the p-value using Benjamini-Hochberg (BH) correction [4].

5. When the adjusted p-value is found to be significant, i.e., less than the specified α, then reject H_0, else do not reject H_0. Based on the decision and the chosen severity requirement, S, obtain the supported δ.

The significance level α, the desired severity, $S \in [0, 1]$, (same as desired power) and the practically relevant δ_p should be chosen by the user. As the power of the test is usually chosen as 80 percent or 95 percent based on the problem domain, similarly, the recommended severity is chosen at 80 percent or 95 percent and α of 95 percent. Based on the performance metric and the problem domain, to identify if the achieved performance improvement is better than the practical relevance, the δ_p shall be chosen carefully. In case of FE as the metric where the available budget is 100000 FEs, minimum of 100 FE improvement at desired severity can be considered as a practically relevant improvement. In case of CPU time as the metric, depending on the application, meaningful time can be chosen. E.g., for an optimization algorithm implemented in an autonomous car, this value can be in centiseconds, and for offline scheduling problems, it can be in several minutes to hours. The supported δ obtained from the algorithm gives a measure of change in statistics required to achieve the expected severity. The rounded-down value of the ratio of δ and δ_p provides the goal difference metric in each match, thereby quantifying the size of a win or a loss.

The *points* scored for each algorithm are similar to the football league scoring system and are obtained based on the decision of the bootstrapped hypothesis testing, the statistical significance of the result, i.e. supported δ at desired severity and the practical relevance of the win or loss, whether, supported $\delta < \delta_p$ or $\delta > \delta_p$. The *goal difference* measures how much the performance is better/worse in terms of practical relevance. The *points* and *goal difference* are calculated as

$$Outcome = \begin{cases} points\text{=}3,\ \text{GD=}\lfloor \delta/\delta_p \rfloor > 0, \text{ if Reject } H_0 \text{ and} & \text{if } \delta > \delta_p, \\ points\text{=}1,\ \text{GD=}\lfloor \delta/\delta_p \rfloor = 0, \text{ if Reject } H_0 \text{ and} & \text{if } \delta < \delta_p, \\ points\text{=}0,\ \text{GD=}\lfloor \delta/\delta_p \rfloor \leq 0, \text{ if not-Reject } H_0. \end{cases}$$

Upon completion of all pairwise comparisons in $\mathcal{C}$, the cumulative *points* and the cumulative goal differences are calculated and the algorithms are ranked based on the *points*. In addition, the mean and standard deviations of the *points* for each algorithms among all functions can be obtained. The points awarded to algorithms that performed well with practical significance are weighted more than twice (three times to be precise) the points awarded to algorithms that performed only statistically significant. This takes into account the fact that practical significance is more relevant in real-world applications and therefore provides a better estimate of the overall performance of the algorithms. On the other hand, weighting practical significance more than three times may skew the results more towards practical significance and therefore statistical significance will have less influence on the final result.

3.3 Properties of the Ranking Scheme

The proposed ranking scheme demonstrates a weak order since it satisfies the comparability, reflexivity, transitivity, anti symmetry properties and it allows for ties, if goal difference does not differentiate the algorithms [2]. Let us define a relation $R(A_i,A_j)$ such that for any two algorithms A_i and A_j in $\mathcal{A}$, P_i, P_j denote the points and GD_i, GD_j denote the goal difference of algorithm A_i and A_j respectively. Note that the symbols $>,<,=$ represent that one algorithm is ranked higher than, lower than, equal to the other algorithm respectively. Then relation $R(A_i,A_j)$ between two algorithms can be formally defined as

$$R(A_i,A_j) = \begin{cases} A_i > A_j, \text{ if } (\ P_i > P_j) \text{ or } (P_i = P_j \text{ and } GD_i > GD_j), \\ A_j > A_i, \text{ if } (P_j > P_i) \text{ or } (P_j = P_i \text{ and } GD_j > GD_i), \\ A_i = A_j, \text{ if } P_i = P_j \text{ and } GD_i = GD_j. \end{cases}$$

Comparability: Every pair of algorithms $(A_i,A_j)_l \in \mathcal{C}$ are comparable by the definition of $R(A_i,A_j)$

Reflexivity: For every algorithm compared, it is equivalent to itself. *Outcome* gives unique value associated with each algorithm. For any $A_i \in \mathcal{A}$,

$$A_i = A_i, \text{ since } P_i = P_i \text{ and } GD_i = GD_i$$

Transitivity: Since the ranking scheme is based on total *Outcome*, for any $A_i, A_j, A_k \in \mathcal{A}$, $A_i > A_j$ and $A_j > A_k$, then $A_i > A_k$ because of the following.

$$\text{if } P_i > P_j \text{ and } P_j > P_k, \text{ then } P_i > P_k$$

$$\text{if } (P_i = P_j \text{ and } GD_i > GD_j) \text{ and } (P_j = P_k \text{ and } GD_j > GD_k)$$

$$\text{then } P_i = P_k \text{ and } GD_i > GD_k.$$

Anti symmetry: The anti-symmetry property is defined by the following relationship.

$$\text{if } A_i > A_j \text{ and } A_j > A_i \implies A_i \simeq A_j$$

This can be proven directly from the definition of $R(A_i, A_j)$ because the condition can be satisfied only when the algorithms are ranked equal.

4 Case study

We compare the expected run time metrics of three hyper parameter optimization techniques along with (1+1)EA and grid search for a family of genetic algorithms on PBO Suite of 25 problems obtained from [21]. The 25 PBO functions include from onemax, leadingones, a linear function with harmonic weights, various W-model-transformes of onemax and leadingones, low autocorrelation binary sequences, ising models, maximum independent vertex set, N-queens problems. The compared HPO techniques include Irace [13], a mixed-integer parallel efficient global optimization [17], the mixed-integer evolution strategy [12].

As (1+1)EA has shown good performance for PBO in [8] it is considered a baseline. The goal is analysing the impact of mutation, crossover, and its combination on a family of $(\mu + \lambda)$ GA algorithms, which results in four tuning parameters: Parent population size, $\mu \in [100]$, Offspring population size ,$\lambda \in [100]$, mutation rate, $P_m \in [.005, .5]$, cross over probability $P_c \in [0, 1]$. Each of the HPO techniques is allocated a budget of 5000 target runs, where each target run refers to 10 independent runs of the $(\mu + \lambda)$ GA configuration suggestion by the HPO techniques. Two different performance metrics are

considered, namely, minimizing the expected runtime (ERT) and maximizing the Area under the empirical CDF curve of running times (AUC). This results in 9 algorithms to be compared: (1+1)EA, GS.AUC, GS.ERT, Irace.AUC, Irace.ERT, MIES.AUC, MIES.ERT, MIP.EGO.AUC, MIP.EGO.ERT. The $(\mu + \lambda)$ GA configurations provided by each of the HPO technique are evaluated for a budget of 50000 function evaluations and the ERT values are obtained for the PBO problems with respect to the targets defined in Table 1 in [21]. And for AUC, set of 100 equally spaced targets ranging from 0 to the targets defined in Table 1 in [21] is considered. Upon identification of the best tuning parameters by each HPO technique, 100 independent runs for these tuned settings are performed and used for further analysis. For each of the algorithm, results of these 100 runs at desired target defined in Table 1 in [21] is used as performance data of this case study and is obtained using IOHanalyzer [19] as explained in [21]. In all runs, values are capped at the budget 50000 function evaluations if the algorithm cannot find the target. The experimental setup for our ranking scheme is as follows: the significance level α is chosen as 0.05, and the desired severity is at recommended value of 0.8. Since the function evaluation is the performance metric in our fixed-target perspective and the total allocated budget is 50000 function evaluations, the practically significant performance improvement, δ_p, can be chosen as minimum of 500 function evaluations. The re-sample size n_b is chosen as 10000. The classical bootstrapped-based hypothesis testing (HT) is also performed for same α and n_b. Each algorithm scores one point for the decision RejectH_0 or zero otherwise. The resulting ranking results are shown in Table 1.

Without going into the details of the performance of the algorithms for each function, we can understand the quantitative performance of each algorithm from the table provided. For example, (1+1)EA clearly outperforms all the other algorithms and tops the table. However, looking at the GD metric, it can be observed that (1+1)EA has performed really poorly for the cases where it failed to win leading to a big negative goal difference at the end. Though MIES.ERT, MIES.AUC, Irace.AUC lags behind (1+1)EA in the *points*, they have not performed really poorly for the cases it lost against other algorithms. If one would like to minimize the worst-case scenarios to achieve robustness, MIES.ERT can be chosen to be applied instead of (1+1)EA. In this case, we

may not achieve the optimum the fastest. However, the results will not be the slowest for some classes of problems either.

Also, our scheme does not suffer from the problem of aggregating results in ranking as function-wise ranking metrics for each algorithm are analyzed as explained in Figure 2. The (1+1)EA algorithm was outperforming MIES.ERT for functions 6,7, 15-17, 19-21 and 23 as shown in Figure 2. However, the positive GD in these functions was very low. For functions 8-10,14,18, and 24, where the (1+1)EA could not outperform, the GD was negative. The MIES.ERT was able to outperform (1+1)EA in functions 2,3,8-14,18,24, and 25 with positive GD for the majority of the functions. The worst GD for MIES.ERT is approx. -100 for function 6 compared to the value of approx. -580 for function 14 in the case of (1+1)EA. This helps us make informed decisions for a given application. A complete picture of which algorithm was best/worst for which set of functions is made available, which is a major strength of the scheme. Considering page restrictions, function-wise ranking metrics is discussed only for the top 2 algorithms. However, the results for all other algorithms are made available online in [6].

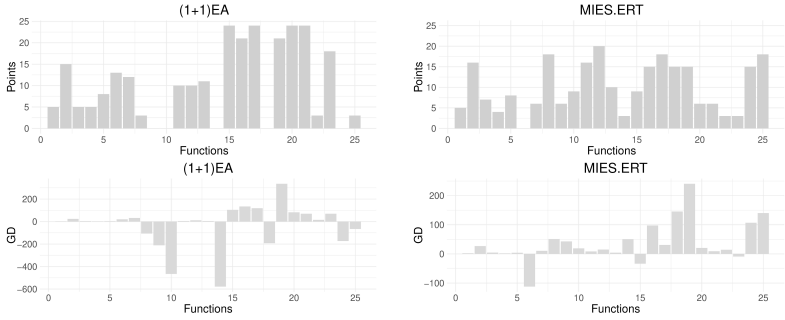


Figure 2: Function-wise ranking metrics of (1+1)EA and MIES-ERT Algorithms

Considering the average mean, median and SD statistics of the *points*, MIES.ERT, MIES.AUC and Irace.AUC exhibits similar performances. Also, the performances of Irace.ERT, MIP.EGO.ERT and MIP.EGO.AUC are comparable. The ranking produced by the classical boostrapped based HT is the same with the only exception of MIP.EGO.AUC outperforming MIP.EGO.ERT by one point. It is also evident that the performances of both algorithms in our proposed ranking

scheme are comparable. Again, this highlights the importance of *GD* metric, which clearly showcases a very high positive goal difference for MIP.EGO.ERT and very large negative goal difference for MIP.EGO.AUC. This explains the order that is provided by the proposed algorithm which gives a larger weight for practical significance (i.e., 3 *points*).

Table 1: Proposed Ranking Scheme Results(S=0.8,δ_p=500) vs classical bootstrapped-based HT. The *points* and GD obtained are the cumulative *points* and GD obtained by each algorithm for all the 25 PBO problems. The position change (PC) in ranking positions is indicated for classical HT with ($\uparrow$, $\downarrow$).

Algorithm	Proposed Ranking		classical HT	
	points	Goal Difference	*points*	PC
1+1 EA	259	-755	107	-
MIES.ERT	251	889	103	-
MIES.AUC	240	899	100	-
Irace.AUC	227	804	89	-
Irace.ERT	169	-207	67	-
MIP.EGO.ERT	161	368	59	$\downarrow 1$
MIP.EGO.AUC	144	-788	60	$\uparrow 1$
GS.ERT	113	-544	41	-
GS.AUC	91	-820	35	-

4.1 Sensitivity analysis of δ_p and S

The sensitivity of input parameters on the ranking solutions are validated at desired severity levels of $S = 0.5, 0.65, 0.8, 0.95$. Similarly, the practically relevant function evaluation is evaluated for a very wide range of $\delta_p = 50, 100, 250$, and the results of the ranking scheme for the resulting 16 experiments are compared. Table 2 presents the influence of severity on the algorithm ranking for δ_p of 500. As expected, as the severity values increase from 50% to 95%, the test becomes more stringent and this is evident in the decreasing trend of *points*. The order of the algorithm rankings is consistent even for various values of severity. Note, however the magnitude of the change is a function of statistics of the solution of the algorithms and cannot be adjudged prior to the experiments. Considering page limitations the detailed results of the influence of δ_p parameter

Table 2: Sensitivity analysis of parameter severity in Ranking Results (δ_p=500). Given a δ_p, the *points* achieved by each algorithm decreases monotonically with respect to the increase in severity. As severity increases, the algorithms that pass a more stringent hypothesis test decreases and accordingly scores less number of *points*. However, no prior statement can be given for the trend in GD for this case because it is a function of δ_p. It is important to note that the sequence of the rankings of the algorithms remains unaltered with various severity levels.

Algorithm	*S* 50%		*S* 65%		*S* 95%	
	points	GD	*points*	GD	*points*	GD
1+1 EA	271	-855	263	-806	249	-669
MIES.ERT	257	952	251	926	241	806
MIES.AUC	246	949	242	930	232	836
Irace.AUC	229	864	227	834	223	736
Irace.ERT	171	-205	171	-205	167	-224
MIP.EGO.ERT	165	344	161	355	159	375
MIP.EGO.AUC	148	-934	146	-865	142	-673
GS.ERT	113	-608	113	-578	109	-494
GS.AUC	95	-975	93	-903	91	-693

on the outcome of the rankings is published in [6]. The order of the algorithm rankings remains consistent for δ_p ranging from 100 to 500. Only when δ_p is 50, the MIP.EGO.AUC secures 178 *points* with -7877 GD and is ahead of MIP.EGO.ERT which secured 177 *points* with 3685 GD. However, this is not a significant improvement for MIP.EGO.AUC, which is evident with the highly negative goal differences. It is to be noted that choosing extreme values for δ_p might influence the sequence of ranking and that is intentional. However, the comparison for all algorithms is performed using the same δ_p and the relative performance improvement is obtained. Even in the existing ranking schemes, the practically relevant improvement parameter is defined as a user-defined variable [9, 10, 18], and extreme choice of this parameter will alter the results. Considering the page restrictions, only certain experiments are presented in Table 2. However, the resulting order of the rankings remained unaltered for the other set of experiments as published in [6]. In addition, repeating the same experiment for several times always produced the same results, convincing of the robust and stable outcome of the proposed ranking scheme. The CRS4EAs ranking scheme as proposed in [18], is used to compare the results obtained by our proposed ranking scheme. The results of the same 9 algorithms for 25

Table 3: Results of CRS4EAs ranking scheme for 2 random runs but for the same input settings. The algorithms are ranked as per the proposed ranking scheme and the position change (PC) in rankings based on CRS4EAs is indicated.

Algorithm	Trial 1				Trial 2			
	R	RD	σ	PC	R	RD	σ	PC
1+1 EA	1586	12.7	0.04	-	1578	12.7	0.04	$\downarrow$1
MIES.ERT	1569	13.5	0.04	$\downarrow$1	1574	13.8	0.04	$\downarrow$1
MIES.AUC	1586	14.9	0.06	$\uparrow$1	1604	14	0.05	$\uparrow$2
Irace.AUC	1563	13.1	0.04	-	1559	14.7	0.05	-
Irace.ERT	1457	13.5	0.04	$\downarrow$2	1451	12.7	0.04	$\downarrow$2
MIP.EGO.ERT	1479	12.8	0.04	$\uparrow$1	1495	13.5	0.04	$\uparrow$1
MIP.EGO.AUC	1473	13.4	0.04	$\uparrow$ 1	1480	13.1	0.04	$\uparrow$ 1
GS.ERT	1377	14.2	0.05	$\downarrow$ 1	1371	14.2	0.05	$\downarrow$ 1
GS.AUC	1410	13.1	0.04	$\uparrow$ 1	1400	13.9	0.04	$\uparrow$ 1

PBO problems are provided as input and in total 5000 games were played. The implementation available in [19] is used for the experiments with the default suggested input parameter settings. The results are shown in Table 3. It is evident that for the same input, the resulting ranking and the ratings keep varying when the same experiment is repeated. Hence, consistent ranking is not observed even with the same inputs.

5 Summary and Outlook

A user-friendly novel ranking scheme based on football league system is proposed that takes into account the statistical significance, the practical significance, and the magnitude of the win or loss of the compared algorithms to determine the final ranking of the algorithms. The proposed scheme has the advantage that the order of comparison has no impact on the results along with there is no necessity of knowing the prior statistics of the compared algorithms. Since the practical significance is used separately from hypothesis testing, the resulting scheme does not make the test more conservative. The proposed scheme shows potential also for comparing machine learning, artificial intelligent(AI), and explainable AI algorithms. For didactic purposes, the HT is

formulated considering the mean performance differences among algorithms. Nevertheless, the scheme can be used to compare the median and related performance measures as well.

References

[1] Bartz-Beielstein, T., Doerr, C., Bossek, J., Chandrasekaran, S., Eftimov, T., Fischbach, A., Kerschke, P., Lopez-Ibanez, M., Malan, K.M., Moore, J.H., et al.: Benchmarking in optimization: Best practice and open issues. arXiv preprint arXiv:2007.03488 (2020)

[2] Bartz-Beielstein, T., Mersmann, O., Chandrasekaran, S.: Ranking and result aggregation. In: Bartz, E., Bartz-Beielstein, T., Zaefferer, M., Mersmann, O. (eds.) Hyperparameter Tuning for Machine and Deep Learning with R: A Practical Guide, chap. 5, pp. 121–161. Springer Nature (2023)

[3] Benavoli, A., Corani, G., Demšar, J., Zaffalon, M.: Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis. The Journal of Machine Learning Research **18**(1), 2653–2688 (2017)

[4] Benjamini, Y., Hochberg, Y.: Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. Journal of the Royal Statistical Society: Series B (Methodological) **57**(1), 289–300 (12 2018).

[5] Calvo, B., Shir, O.M., Ceberio, J., Doerr, C., Wang, H., Bäck, T., Lozano, J.A.: Bayesian performance analysis for black-box optimization benchmarking. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. pp. 1789–1797 (2019)

[6] Chandrasekaran, S.: Data and results of paper titled: A novel ranking scheme for the performance analysis of stochastic optimization algorithms using the principles of severity. `https://doi.org/10.5281/zenodo.10969073`

[7] Chandrasekaran, S., Bartz-Beielstein, T.: A robust statistical framework for the analysis of the performances of stochastic optimization algorithms using the principles of severity. In: International Conference on the Applications of Evolutionary Computation (Part of EvoStar). pp. 426–441. Springer (2023)

[8] Doerr, C., Ye, F., Horesh, N., Wang, H., Shir, O.M., Bäck, T.: Benchmarking discrete optimization heuristics with iohprofiler. Applied Soft Computing **88**, 106027 (2020)

[9] Eftimov, T., Korošec, P.: Identifying practical significance through statistical comparison of meta-heuristic stochastic optimization algorithms. Applied Soft Computing **85**, 105862 (2019)

[10] Eftimov, T., Korošec, P.: A novel statistical approach for comparing meta-heuristic stochastic optimization algorithms according to the distribution of solutions in the search space. Information Sciences **489**, 255–273 (2019)

[11] Gelman, A.: Objections to Bayesian statistics. Bayesian Analysis **3**(3), 445 – 449 (2008)

[12] Li, R., Emmerich, M.T., Eggermont, J., Bäck, T., Schütz, M., Dijkstra, J., Reiber, J.H.: Mixed integer evolution strategies for parameter optimization. Evolutionary computation **21**(1), 29–64 (2013)

[13] López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L.P., Birattari, M., Stützle, T.: The irace package: Iterated racing for automatic algorithm configuration. Operations Research Perspectives **3**, 43–58 (2016)

[14] Mayo, D.G., Spanos, A.: Severe testing as a basic concept in a neyman-pearson philosophy of induction. The British Journal for the Philosophy of Science **57**(2), 323–357 (2006)

[15] Neyman, J., Pearson, E.S.: On the use and interpretation of certain test criteria for purposes of statistical inference: Part i. Biometrika pp. 175–240 (1928)

[16] Rojas-Delgado, J., Ceberio, J., Calvo, B., Lozano, J.A.: Bayesian performance analysis for algorithm ranking comparison. IEEE Transactions on Evolutionary Computation **26**(6), 1281–1292 (2022)

[17] van Stein, B., Wang, H., Bäck, T.: Automatic configuration of deep neural networks with parallel efficient global optimization. In: 2019 International Joint Conference on Neural Networks (IJCNN). pp. 1–7. IEEE (2019)

[18] Veček, N., Mernik, M., Črepinšek, M.: A chess rating system for evolutionary algorithms: A new method for the comparison and ranking of evolutionary algorithms. Information Sciences **277**, 656–679 (2014)

[19] Wang, H., Vermetten, D., Ye, F., Doerr, C., Bäck, T.: Iohanalyzer: Detailed performance analyses for iterative optimization heuristics. ACM Trans. Evol. Learn. Optim. **2**(1) (apr 2022).

[20] Wasserstein, R.L., Lazar, N.A.: The ASA's statement on p-values: Context, process, and purpose. The American Statistician **70**(2), 129–133 (2016)

[21] Ye, F., Doerr, C., Wang, H., Bäck, T.: Automated configuration of genetic algorithms by tuning for anytime performance. IEEE Transactions on Evolutionary Computation **26**(6), 1526–1538 (2022)