Bayesian Performance Analysis for Algorithm Ranking Comparison

Jairo Rojas-Delgado, Josu Ceberio, Borja Calvo, and Jose A. Lozano, *Fellow, IEEE*

***Abstract*—In the field of optimization and machine learning, the statistical assessment of results has played a key role in conducting algorithmic performance comparisons. Classically, null hypothesis statistical tests have been used. However, recently, alternatives based on Bayesian statistics have shown great potential in complex scenarios, especially when quantifying the uncertainty in the comparison. In this work, we delve deep into the Bayesian statistical assessment of experimental results by proposing a framework for the analysis of several algorithms on several problems/instances. To this end, experimental results are transformed to their corresponding rankings of algorithms, assuming that these rankings have been generated by a probability distribution (defined on permutation spaces). From the set of rankings, we estimate the posterior distribution of the parameters of the studied probability models, and several inferences concerning the analysis of the results are examined. Particularly, we study questions related to the probability of having one algorithm in the first position of the ranking or the probability that two algorithms are in the same relative position in the ranking. Not limited to that, the assumptions, strengths, and weaknesses of the models in each case are studied. To help other researchers to make use of this kind of analysis, we provide a Python package and source code implementation at https://zenodo.org/record/6320599.**

***Index Terms*—Bayesian inference, benchmarking, evolutionary algorithms, probabilistic models on permutation spaces.**

I. Introduction

THE ANALYSIS of empirical results is of critical importance in many scientific disciplines. According to López-Ibáñez *et al.* [1], until a mathematical proof is discovered, the ability to reach consistent conclusions, through experimental repetition performed by other researchers, is the only way a research community can reach a consensus.

Usually, empirical results are surrounded by uncertainty and, therefore, we need to handle this uncertainty to extract sound conclusions. Classically, this uncertainty has been addressed with the use of statistical tests. This approach focuses on testing a certain hypothesis and deciding whether there is enough statistical evidence in the results to reject that hypothesis. Most likely, the results will show some degree of difference in the performance of two given algorithms, and the test is used to reject the null hypothesis.

Null hypothesis statistical tests have several weaknesses that were recently highlighted by Benavoli *et al.* [2]. Among many issues, arguably, the most relevant aspect is what Benavoli *et al.* called the *black and white thinking*. Certainly, the statistical test provides a reference of the uncertainty, the so-called p-value, but this value is not easy to interpret, as it mixes *the magnitude of the difference* and *the sample size* [3], [4], [5]. Rather than interpreting this value, usually, a threshold is set (0.05 by convention) to decide whether the differences are due to chance or not, hence the binary nature of the tests.

Bayesian statistics provide an interesting alternative to the classical statistical test approach, as they naturally handle the uncertainty remaining after observing the results of the experimentation. This is achieved by updating our prior belief in whatever aspect we are interested in (e.g., the magnitude of the differences between two algorithms) with the evidence. The Bayesian approach broadens the possible analysis, as we are not limited to just those statistics for which a p-value can be computed. In this article, we propose a methodology to analyze the results from a ranking perspective.

We are interested in comparing the performance of a set of algorithms $A = \{A_1, \ldots, A_n\}$ when solving a set of problem instances $I = \{I_1, \ldots, I_p\}$. The set of problem instances can be as small as a single problem instance. We record a score $F_{i,j} = f(A_i, I_j)$ for each algorithm and problem instance. We consider that $F_{i,k} < F_{j,k}$ means that the ith algorithm outperforms the jth algorithm on the kth problem instance. In practice, we usually run an algorithm in a problem instance several times obtaining a score for each repetition: $F_{i,j,1}, \ldots, F_{i,j,r}$ where r is the number of repetitions.

In computer science and especially when comparing the results of evolutionary algorithms, we usually compare a set of algorithms in a set of problem instances and draw conclusions based on such performance analysis. The way we proceed with this comparison has deep implications that are often not considered. First of all, in many cases, we do not have access to

Manuscript received 15 September 2021; revised 19 March 2022 and 26 June 2022; accepted 9 September 2022. Date of publication 20 September 2022; date of current version 1 December 2022. This work was supported in part by the Spanish Ministry of Science and Innovation under Project PID2019-104933GB-10/AEI/10.13039/501100011033 and Project PID2019-106453GAI00/AEI/10.13039/501100011033 and (BCAM Severo Ochoa Accreditation) under Grant SEV-2017-0718; and in part by the Basque Government through the BERC Program 2022-2025 and through the Elkartek Program under Grant KK.2020/00049, Grant KK.2021/00091, and Grant IT1504-22. *(Corresponding author: Jairo Rojas-Delgado.)*

Jairo Rojas-Delgado is with the Machine Learning Group, Basque Center for Applied Mathematics, 48009 Bilbao, Spain.

Josu Ceberio and Borja Calvo are with the Intelligent Systems Group, University of the Basque Country (UPV/EHU), 20018 Donostia-San Sebastian, Spain.

Jose A. Lozano is with the Machine Learning Group, Basque Center for Applied Mathematics, 48009 Bilbao, Spain, and also with the Intelligent Systems Group, University of the Basque Country (UPV/EHU), 20018 Donostia-San Sebastian, Spain.

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TEVC.2022.3208110.

Digital Object Identifier 10.1109/TEVC.2022.3208110

the entire set of problem instances of interest, nor do we know the entire set of algorithms to compare. Moreover, the algorithms are typically stochastic, meaning that each time they are run they may provide a different result. All these aspects have to do with the definition of the population from which we are doing inference, and this population can be different in different analyses. It is worth noting that we are doing inference, and that has very important implications. Specifically, we will try to draw conclusions about a certain population by analyzing a sample from that population.

In a real-life situation, the first step would be to decide which is our target population. There are different alternatives here that involve determining the set of problem instances, the set of algorithms, the way different sources of variance are considered together or not and, in general, the conditions under which we run the algorithms in the problem instances. The important consideration to keep in mind is that the conclusions we obtain from our analysis only apply to the population we are considering and any further generalization to a different population is risky. From a general perspective, in our work, we focus our attention on populations that involve paired and noncomparable comparisons due to their practical relevance in the comparison of evolutionary algorithms.

Paired comparisons mean that the score of a given algorithm obtained in a particular problem instance is paired to the score obtained by the other algorithms in that particular problem instance, but it is not paired to other problem instances. In other words, deriving conclusions from $F_{i,k} < F_{j,h}$ for $k \neq h$ should not be included in the analysis. For example, we should not compare the score of algorithm A_i when solving a small problem instance with the score of algorithm A_j in a different and larger problem instance. The scenario in which we conduct several repetitions of the algorithms in the same problem instance deserves special consideration. In this case, we could consider crossing the scores obtained from the different repetitions within the same problem instance.

Noncomparable observations mean that the score of a specific algorithm on a problem instance is not comparable to the score of the same algorithm on a different problem instance, for example, because the score is on a different scale. This creates difficulties with the use of simple statistics, such as the mean score of the algorithm in the set of problem instances, to summarize and compare the different algorithms.

In such a scenario, paired and noncomparable observations, ranking data seems a natural choice to model the performance of the algorithms. The performance of a single run of the different algorithms on the kth problem instance is represented by a permutation of n-items $\pi \in S_n$, where S_n is the set of permutations of n-items such as $F_{\pi(1),k,j} < \cdots < F_{\pi(n),k,j}$. Therefore, instead of dealing with the raw score data $F_{i,j,k}$, we deal with a set of permutations of n-items $S = \{\pi_1, \pi_2, \ldots, \pi_{pr}\}$, one permutation per instance-repetition.

The comparison and benchmarking of evolutionary computation methods and other optimization algorithms is a wide and active area of research [6], [7]. Perhaps, the first paper for the Bayesian estimation of ranking models in evolutionary computing was introduced by Calvo *et al.* [8] in which the authors consider a ranking model known as Plackett–Luce (PL). The interpretation of the resulting ranking model parameters and their associated probability distributions is used to draw conclusions on the performance of the algorithms. Specifically, Calvo *et al.* studied the probability that one algorithm is the best. More recently, Mattos *et al.* [9] discussed several practical aspects related to the Bayesian data analysis, such as the interpretation of posterior summaries, the need to check for the Monte Carlo sampling convergence and different kinds of sensitivity analysis that can be performed. In a similar research direction, Carrasco *et al.* [6] surveyed recent statistical analyses for the comparison of evolutionary algorithms, including several Bayesian tests and how to address some of the criticisms of null hypothesis statistical tests.

We extend the previous work of Calvo *et al.* [8] by considering additional probabilistic models on permutation spaces for Bayesian inference. We explore the richness of additional posterior summaries that can be considered by the proposed ranking models beyond the probability that one is the best. In the same spirit of the previous works regarding the importance of not considering the Bayesian performance analysis as a black-box tool, we carefully review the assumptions the proposed models make, their properties and the implications of such assumptions and properties in our analysis. We study five properties of the proposed probabilistic models on permutation spaces previously discussed by Critchlow *et al.* [10]: 1) label invariance; 2) reversibility; 3) L-decomposability; 4) strong unimodality; and 5) complete consensus.

The remainder of this manuscript is organized as follows. In Section II, we describe the ranking models studied in this work. In Section III we describe how, through careful interpretation of the ranking model parameters when conducting Bayesian inference, we can draw and report conclusions on the performance of the algorithms. In Section IV, we develop a case study with synthetic data and real data in which we compare the performance of several evolutionary algorithms. In Section V, we provide some general guidelines and an outlook for future work on this particular topic.

II. Probability Models on Permutation Spaces

In this section, we study several probability models on permutation spaces for the Bayesian inference. We review the definition, main assumptions, and properties of such models. This is not an exhaustive review and the interested reader will find in the literature other models that have not been considered here.

A. Bradley–Terry Model

The Bradley–Terry (BT) model dates back to at least 1929 and has applications in a broad range of problems [11]. This model is used in a situation in which the items to compare (algorithms in our context) are repeatedly compared with one another in pairs such as

$$\Pr\left[A_i \text{ outperforms } A_j\right] = \frac{\theta_i}{\theta_i + \theta_j} \tag{1}$$

where θ_i is a positive-valued parameter associated with algorithm A_i. Hence, we are dealing with n parameters.

Considering this model, a ranking can be generated by sampling from the distribution in (1), where, for every pair, a preference relation is obtained, i.e., either A_i is preferred to A_j or vice versa. If there are no circular triads (e.g., A_i is preferred to A_j, A_j is preferred to A_k, and A_k is preferred to A_i) a ranking can be produced. Consequently, the probability of observing a ranking is given as follows:

$$\Pr[\pi|\boldsymbol{\theta}] = \frac{1}{\psi(\boldsymbol{\theta})} \prod_{i=1}^{n-1} \left[\theta_{\pi(i)}\right]^{n-i} \tag{2}$$

where $\psi(\boldsymbol{\theta})$ is a constant, whose value does not depend on π, chosen to make the probabilities sum to 1 and $\boldsymbol{\theta} \in \Omega_{\boldsymbol{\theta}} \subset \mathbb{R}^n$. For completeness

$$\psi(\boldsymbol{\theta}) = \sum_{\pi \in S_n} \prod_{i=1}^{n-1} \left[\theta_{\pi(i)}\right]^{n-i}. \tag{3}$$

The BT model has been extended to allow for comparisons among more than two items at once. The generalization to comparisons of any number of items is known as the PL model.

B. Plackett–Luce Model

When considering the PL model, the Luce axiom states that the probability of item A_i outperforming the other items in the set A is

$$\Pr\left[A_i \text{ outperforms } A_j \quad \forall A_j \in A\right] = \frac{\theta_i}{\sum_{A_j \in A} \theta_j} \tag{4}$$

such as $A_i \notin A$ and θ_i is a real positive parameter related to the goodness of the ith item. The probability of observing a given ranking under the PL model is given by

$$\Pr[\pi|\boldsymbol{\theta}] = \prod_{i=1}^{n} \frac{\theta_{\pi(i)}}{\sum_{j=i}^{n} \theta_{\pi(j)}} \tag{5}$$

where $\boldsymbol{\theta} \in \Omega_{\boldsymbol{\theta}} \subset \mathbb{R}^n$. Moreover, by restricting $\sum_{i=1}^{n} \theta_i = 1$, the model parameters can be interpreted as the probability of each item being the top ranked, i.e., $\Pr[\pi(1) = i] = \theta_i$.

C. Mallows Model

The Mallows model (MM) is one of the preferred distributions to model ranking data. It belongs to the location-scale family since it is parametrized by a location parameter (also known as central ranking) $\pi_0 \in S_n$ and a non-negative scale (also known as dispersion) parameter, β.

The location parameter is the consensus ranking of the distribution. The probability of any other permutation decreases exponentially with its distance to π_0. In our work, Kendall's-tau metric has been chosen to measure the distance between rankings.[1] The dispersion parameter controls the variance of this decay. Considering the MM, the probability of observing a given ranking is given by

$$\Pr[\pi|\boldsymbol{\theta} = (\pi_0, \beta)] = \frac{\exp(-\beta d(\pi_0, \pi))}{\psi_n(\beta)} \tag{6}$$

[1]Other distance metrics can be consulted in [12].

where $\boldsymbol{\theta} \in \Omega_{\boldsymbol{\theta}} \subset S_n \times \mathbb{R}$ and $d(.,.)$ is the Kendall-tau distance between two permutations, i.e.,

$$d(\sigma, \sigma') = \sum_{i<j} \mathbb{I}\left[(\sigma(i) - \sigma(j)) \cdot \left(\sigma'(i) - \sigma'(j)\right) < 0\right] \tag{7}$$

where $\mathbb{I}[\,\cdot\,]$ denotes the indicator function and $\psi_n(\beta)$ is a normalization constant, which in the case of the Kendall-tau distance is given by

$$\psi_n(\beta) = \prod_{i=1}^{n-1} \frac{1 - \exp(-\beta(n-i+1))}{1 - \exp(-\beta).} \tag{8}$$

III. Bayesian Models for Algorithm Ranking Comparison

In this section, we briefly review how the Bayesian inference can be used for algorithm performance analysis and describe the use of probability models on permutation spaces for this task. We carefully describe how to define the related likelihood functions and prior distributions within the Bayesian inference framework, and analyze whether special modifications are required to come up with a Markov Chain Monte Carlo approach to sample from the posterior distribution.

Bayesian inference is a method of statistical inference in which Bayes' theorem is used to update the distribution of some parameters as more data or information become available. Briefly, in the Bayesian inference, we are interested in the posterior distribution given by

$$\Pr[\boldsymbol{\theta}|S] = \frac{\Pr[S|\boldsymbol{\theta}] \cdot \Pr[\boldsymbol{\theta}]}{\Pr[S]}. \tag{9}$$

The main components for Bayesian inference are listed as follows.

1) The likelihood function $Pr[S|\boldsymbol{\theta}]$, which represents the probability of observing the data S assuming that a given set of parameters $\boldsymbol{\theta}$ explains such data.
2) The prior probability $\Pr[\boldsymbol{\theta}]$, representing the probability of the parameters before observing any data.
3) The data probability $\Pr[S]$, representing the probability of observing the data independently of the parameters.

In our context, data refers to the ranking data $S = \{\pi_1, \ldots, \pi_p\}$ where $\pi_i \in S_n$ and parameters refer to the ranking probability model parameters, which we denote as $\boldsymbol{\theta} \in \Omega_{\boldsymbol{\theta}}$. Formally, we have a probability mass function with support on S_n for the data which is specified by the ranking probabilistic model. If we assume that the ranking data is a sample of i.i.d. permutations, then the likelihood function is given by

$$\Pr[S|\boldsymbol{\theta}] = \prod_{\pi \in S} \Pr[\pi|\boldsymbol{\theta}] \tag{10}$$

where $\Pr[\pi|\boldsymbol{\theta}]$ denotes such a probability mass function. The prior distribution encodes any preference for the ranking model parameters $\boldsymbol{\theta}$. Formally, the prior distribution is given by a probability distribution with support $\Omega_{\boldsymbol{\theta}}$ denoted as follows:

$$\Pr[\boldsymbol{\theta}|\boldsymbol{\alpha}] \tag{11}$$

where $\boldsymbol{\alpha}$ parametrizes this probability distribution (also known as hyperparameter). The data probability is the distribution

of the observed data marginalized over the parameters of the model

$$\Pr[S|\boldsymbol{\alpha}] = \int_{\Omega_{\boldsymbol{\theta}}} \Pr[S|\boldsymbol{\theta}] \cdot \Pr[\boldsymbol{\theta}|\boldsymbol{\alpha}] d\boldsymbol{\theta} \tag{12}$$

which does not depend on the parameters. Obtaining this last component is the main challenge to come up with a closed-form expression in the Bayesian inference. Therefore, approximation methods are used, such as Markov Chain Monte Carlo methods, e.g., the Metropolis–Hastings algorithm. With such approximation methods, we can use a function proportional to the posterior instead of the posterior itself, such as

$$\Pr[\boldsymbol{\theta}|S] \propto \Pr[S|\boldsymbol{\theta}] \cdot \Pr[\boldsymbol{\theta}|\boldsymbol{\alpha}]. \tag{13}$$

In the following sections, we study the likelihood functions, prior distributions, and posterior summaries associated with the probability models on permutation spaces.

A. Likelihood Functions

In our specific application of algorithm performance analysis, the likelihood function represents the probability of observing a set of rankings S given some parameters $\boldsymbol{\theta} \in \Omega_{\boldsymbol{\theta}}$. We assume that the rankings in S are i.i.d., which allows us to obtain the likelihood function as described in (10). To complete the definition of the likelihood function for each model, the probability of observing a ranking $\Pr[\pi|\boldsymbol{\theta}]$ is given by each probability model: for the BT model as specified in (2), for the PL model as specified in (5) and for the MM model as specified in (6).

B. Priors Specification

Mattos *et al.* [9] made a distinction between noninformative, weakly informative, and informative priors based on how much information is included in the Bayesian model. In practical settings, choosing between different prior distributions and their hyperparameters should be made based on some previous knowledge (e.g., by making a review of the state-of-the-art and giving more weight to the algorithms that perform better). When no previous knowledge exists, then a noninformative prior should be preferred. It is the role and responsibility of the researcher, the reviewer, and the community to assess if the knowledge encoded in the prior distributions is in line with the state-of-the-art. Nonetheless, the effect of the prior distribution on the posterior distribution diminishes as the amount of data grows [13]. However, this is just an asymptotic result and it does not guarantee that an arbitrary prior will give a consistent Bayesian estimate of the unknown parameter in all circumstances [14]. In this direction, several recent works have addressed the important issue of whether the posterior distributions derived with distinct priors become very similar if more data is gathered [15], [16].

Beyond such general considerations, we may conduct a sensitivity analysis, in which we corroborate that the results of our analysis are not too different when using different prior distributions. This is an important step to verify the robustness of our analysis which is widely used in related studies, such as Calvo *et al.* [8] and Mattos *et al.* [9]. In the supplementary material of our work, we provide details of how to conduct such sensitivity analysis as an example. Another alternative for modeling prior distributions is to take a hyper prior distribution and perform hierarchical Bayesian inference [2]. We model the prior distributions as follows.

1) *Bradley–Terry:* In the BT model, $\boldsymbol{\theta}$ can be multiplied by an arbitrary positive constant $k > 0$ without affecting the associated probability in (1). Therefore, two-parameter vectors are equivalent if one is a scalar multiple of the other, and, consequently, we can constrain the parameter space to sum one. This allows us to model the prior distribution for $\boldsymbol{\theta}$ using the Dirichlet distribution with $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n), \alpha_i > 0$ concentration parameters.
2) *Plackett–Luce:* The PL model is a Thurstone model in which the rankings are given by the relative ordering of n i.i.d. real random variables $X_1, \ldots, X_n$ such that each variable is Gumbel distributed and only differs in their location parameter [17]. In other words, $X_i \sim g(\mu_i, \beta)$ where $g(\mu_i, \beta)$ is the Gumbel distribution with location parameter μ_i and scale parameter β. Guiver and Snelson [18] showed that given the location parameter of the Gumbel distribution for the ith i.i.d. random variable of the Thurstone model μ_i and a fixed scale parameter β, the PL model parameters relate to the location parameter of the Thurstone model as follows $\theta_i = \exp(\mu_i/\beta)$ and have a Gamma distribution conjugate prior. This has some interesting ramifications in the definition of the prior probability distribution in the Bayesian analysis. Therefore, we consider two prior distributions.
   a) The individual parameters θ_i, modeled using the Gamma distribution [18].
   b) The parameter vector $\boldsymbol{\theta}$, modeled using the Dirichlet distribution [8].
3) *Mallows Model:* For this model, the prior distribution can be modeled by providing a prior for the central ranking and for the dispersion parameter. We use the uniform prior for the central ranking and a truncated exponential prior for the dispersion parameter, as suggested by [19]

$$\pi_0 \sim \Pr[\pi_0] = \frac{1}{n!} \tag{14}$$

$$\beta \sim \Pr[\beta|\lambda, \beta_{\max}] = \frac{\lambda \exp(-\lambda\beta)}{1 - \exp(-\lambda\beta_{\max})} \tag{15}$$

where λ and $\beta_{\max}$ are two hyperparameters.

C. Posterior Summaries

In the Bayesian analysis, we get a posterior distribution of the parameters of the proposed models, i.e., $\Pr[\boldsymbol{\theta}|S]$. With such posterior distribution, we can get point estimates of the parameters of the models (e.g., the mean) and more importantly credible intervals that provide an estimate of the uncertainty.

In general, the posterior distribution of the model parameters is not particularly informative. Therefore, we derive from the posterior distribution of the model parameters, other more informative posterior distributions, such as the posterior distribution of a given algorithm being the best. For example, given one sample of the posterior distribution for the parameters of

the PL model $\boldsymbol{\theta} \sim \Pr[\boldsymbol{\theta}|S]$, we can obtain the probability of algorithm A_i being the best as follows:

$$\Pr[\pi(1) = i] = \theta_i \tag{16}$$

From the derived posterior distributions, we can retrieve again point estimates or credible intervals with an uncertainty estimation quantified in the variance of the resulting posterior distributions.

Particularly, we focus our attention on the following posterior summaries: 1) the probability of an algorithm being ranked the first; 2) the probability of an algorithm outperforming another algorithm; and 3) the probability of an algorithm being in the top-k ranking.

1) The probability of an algorithm A_i being ranked the first is given by the following expression:

$$\Pr[\pi(1) = i] = \sum_{\pi(1)=i} \Pr[\pi|\boldsymbol{\theta}]. \tag{17}$$

2) The probability of a given algorithm A_i outperforming another algorithm A_j is

$$\Pr\left[\pi^{-1}(i) < \pi^{-1}(j)\right] = \sum_{\pi^{-1}(i)<\pi^{-1}(j)} \Pr[\pi|\boldsymbol{\theta}] \tag{18}$$

where π^{-1} is the inverse of the permutation.

3) The probability of an algorithm A_i being in the top-k ranking is

$$\Pr\left[\pi^{-1}(i) \leq k\right] = \sum_{\pi^{-1}(i)\leq k} \Pr[\pi|\boldsymbol{\theta}]. \tag{19}$$

Additional Posterior Summaries: Though in this work we only explore a few interesting questions related to posterior summaries, several others may be of interest to the community and are left for future works, for example:

1) probability that a given algorithm is ranked in a given position other than the first one;
2) given a subset of the algorithms, the probability that a given algorithm is better than all the algorithms in that subset;
3) given two disjoint subsets of the algorithms, the probability that all the algorithms in a subset appear in a ranking before the algorithms in the other subset.

The richness of posterior summaries that can be obtained from the posterior distributions of the ranking model parameters is among their main attractiveness. Simple posterior summaries such as answering binary questions (e.g., is algorithm A_1 better than algorithm A_2) are naturally covered in the scenario of the Bayesian ranking comparison of several algorithms (as it is the particular case of rankings of two elements). Nevertheless, when comparing just two algorithms there are indeed other more specific approaches reported in the literature, such as that proposed by Benavoli *et al.* [2] which considers a rope parameter to account for the cases in which the two algorithms perform similarly.

Computational Complexity: In the naive case, the previously discussed marginal probabilities have a computational complexity of $\mathcal{O}(n!)$ which comes from the large sample space of the probabilistic models on permutation spaces. For some of the proposed models and marginal probabilities of interest, some closed-form expressions may be available. For example, the probability of a given algorithm being ranked the first can be obtained in $\mathcal{O}(1)$ for the PL as stated in (16) and in $\mathcal{O}(n)$ for the MM using the Kendall-tau distance if we consider the result of Collas and Irurozki [20] (Lemma 1) as follows:

$$\Pr[\pi(1) = i] = \exp\left(-\beta\left(\pi_0^{-1}(i) - 1\right)\right)\frac{\psi_{n-1}(\beta)}{\psi_n(\beta)} \tag{20}$$

where $\boldsymbol{\theta} = (\pi_0, \beta) \in S_n \times \mathbb{R}$ are the MM parameters (central permutation and dispersion parameter). However, in cases in which no closed-form expression exists, we may be limited by expensive computations when the number of algorithms to be compared (n) is not small. Exploring different probabilistic models on permutation spaces is in part motivated by the high computational complexity of obtaining some of these marginal probabilities in cases where no closed form expressions are available.

D. Bayesian Analysis: Additional Considerations

In Bayesian analysis, many arguments have been provided regarding different important issues. In this sense, topics such as selecting the number of samples used as burnout in the Markov Chain Monte Carlo sampling, the need to check the convergence of the Monte Carlo chains or how Bayesian analysis provides a quantification of the uncertainty are widely covered in [9]. In this section, we provide additional background on practical issues related to the Bayesian ranking comparison of algorithms beyond such general issues.

Given the scores of a set of algorithms when solving a set of problem instances, we create a set of permutations that represents the rankings of the algorithms. In this section, we describe how to obtain a set of permutations from the scores of the different algorithms while dealing with practical issues such as ties between the algorithms.

Ties Between Algorithms: In practical settings, when comparing the performance of two or more algorithms in the same problem instance and repetition pair, there can be ties, that is, $F_{i,j,k} = F_{i',j,k}$ for $i \neq i'$. We deal with this kind of situation by obtaining several rankings for each problem instance and repetition pair in which ties exist. The number of additional rankings corresponds to all possible ways in which the ties may be resolved in favor of one algorithm or another.

After collecting the permutations from the score data, including the additional permutations coming from resolving ties, we need to consider the bias induced by including more than one permutation from a single problem-instance pair. For example, six permutations generated by a triple-tie should not count the same as six permutations obtained from six different repetitions.

The ideal solution in these cases would be to: 1) take all permutations in which there are no ties and give those a weight of one and 2) take all other permutations in which there are t ties and resolve all ties generating $t!$ permutations while giving those a weight of $1/t!$ The next step would be to consider all the resulting permutations in the Bayesian inference, but

include the weight in the likelihood function, that is

$$\Pr[S|\boldsymbol{\theta}] = \prod_{\pi \in S} w_\pi \Pr[\pi|\boldsymbol{\theta}] \tag{21}$$

where w_π is the weight associated with permutation π and S is the set that contains all the permutations obtained from the scores, including the ones obtained by solving the ties.

In practice, however, when there are many ties, solving all ties this way yields a factorial number of permutations which may be prohibitive. Therefore, we could opt to choose some sample estimator of the likelihood function. In this direction, we could take a sample in which, first, we include all the permutations that do not have ties and give those a weight of one and, second, solve the ties of the remaining permutations in which a number of t ties exists. However, instead of including all the $t!$ resulting permutations, we include a maximum number k of those. If $t! < k$, then, we just include all the permutations, otherwise, we take a uniform subsample from the set of $t!$ permutations and give them a weight of $1/k$.

Naturally, we need some sensible ways of determining the value of k. In this sense, our first recommendation would be to take the largest possible value of k according to practical computational constraints. A second recommendation goes in the direction of conducting a sensitivity analysis to verify if the general overview and conclusions are affected when we vary the number of k after some point.

Additional Models on Permutation Spaces: Other probabilistic models on permutation spaces may be of interest to perform Bayesian analyses of algorithm performance. Considering several other models to this end may be adequate to fit some known assumptions of the data, add more flexibility to the analyses or enable the computation of some posterior summaries more efficiently. Some examples of such probabilistic models on permutation spaces are listed as follows.

1) The generalized MM under different distances: Kendall-tau, Ulam, and Cayley.
2) The weighted MM under the Hamming distance as introduced in [12].

E. Properties, Assumptions, Pros and Cons

Let, $\Pr[\pi] \in \mathcal{P}$ be a probabilistic model where $\pi \in S_n$ and $\Pr[\pi]$ denotes its probability mass function whereas $\mathcal{P}$ denotes a class of such models, usually indexed by a set of parameters. A probabilistic model on permutation spaces can be characterized based on different properties, such as label invariance, reversibility, L-decomposability, strong unimodality, and complete consensus [10].

1) *Label Invariance:* A class of distributions $\mathcal{P}$ is label invariance if for all $\Pr[\pi] \in \mathcal{P}$ and a relabeling permutation $\gamma \in S_n$, there is another probability distribution $\Pr_\gamma \in \mathcal{P}$ such as $\Pr_\gamma[\pi \circ \gamma] = \Pr[\pi]$. Here, $\pi \circ \gamma = \pi(\gamma(1)), \ldots, \pi(\gamma(n))$ is the composition operation.
2) *Reversibility:* A class of distributions $\mathcal{P}$ has the reversibility property if for all $\Pr[\pi] \in \mathcal{P}$, there is another $\Pr_\gamma \in \mathcal{P}$ in which reversing the natural linear ordering of the rankings, i.e., $\gamma \circ \pi$ with $\gamma \in S_n$, $\gamma(i) = n + 1 - i$ yields $\Pr[\pi] = \Pr_\gamma[\gamma \circ \pi]$.

TABLE I
PROPERTIES OF THE PROBABILITY MODELS ON PERMUTATION SPACES

	BT	PL	MM
Label invariance	Yes	Yes	Yes
Reversibility	Yes	No	Yes
L-Decomposability	Yes	Yes	Yes
Strong unimodality	Yes	Yes	Yes
Complete consensus	Yes	Yes	Yes

3) *L-Decomposability:* A model is said to be L-decomposable if the probability of a ranking π can be expressed as $\prod_{i=1}^{n} \Pr[\pi(i)|\{\pi(i+1), \ldots, \pi(n)\}]$, where $\Pr[\pi(i)|\{\pi(i+1), \ldots, \pi(n)\}]$ is the probability that item $\pi(i)$ is better than the items in $\{\pi(i+1), \ldots, \pi(n)\}$.
4) *Strong Unimodality:* A model is said to have the strong unimodality property if given the mode of the distribution $\pi_0 \in S_n$, for each pair of items (i, j), such as $\pi_0(i) < \pi_0(j)$ and any permutation $\pi \in S_n$ in which $\pi(i) = \pi(j) - 1$, $\Pr[\pi] \geq \Pr[\pi \circ \gamma_{i,j}]$ with $\gamma_{i,j} \in S_n, \gamma_{i,j}(i) = j, \gamma_{i,j}(j) = i, \gamma_{i,j}(m) = m$ for all $m \neq i, j$. Here, $\pi \circ \gamma_{i,j}$ is the permutation that agrees with π except that the ranks assigned to items i and j are exchanged.
5) *Complete Consensus:* A model is said to have the complete consensus property if given the mode of the distribution $\pi_0 \in S_n$, for each pair of items (i, j) such as $\pi_0(i) < \pi_0(j)$ and any permutation $\pi \in S_n$ in which $\pi(i) \leq \pi(j)$, $\Pr[\pi] \geq \Pr[\pi \circ \gamma_{i,j}]$ with $\gamma_{i,j} \in S_n$ defined as before. Complete consensus implies strong unimodality.

Table I summarizes the properties of the different probabilistic models in permutation spaces. In general, all models share the same properties except for the PL model, which does not have the reversibility property.

In some cases, these properties are important in determining the correct use of the models when conducting the Bayesian performance analysis for algorithm ranking comparison. In other cases, some of these properties are shared by all models and we highlight how it makes sense in the context of algorithm ranking comparison.

Label Invariance and Reversibility: Label invariance is a property that is shared by the BT, the PL, and the MM. This property ensures that the results of the Bayesian analysis we conduct using these models are invariant to any arbitrary relabeling of the algorithm scores. This is the same as saying that the name of the algorithms or the order in which we compare their scores has no impact on the results of our analysis.

Reversibility is a property that is shared only by the BT and the MM but not by the PL model. When a model has the reversibility property, it means that we can do the same analysis if we rank the algorithms from best to worst or from worst to best, because in either case, the given probability model in permutation spaces can represent both probability distributions.

As an example of the consequences of this, consider the case in which we are interested in two different posterior summaries: 1) the probability of an algorithm being the best, i.e.,

$\Pr[\pi(1) = A_i]$ and 2) the probability of an algorithm being the worst, i.e., $\Pr[\pi(n) = A_i]$. We may be tempted to obtain two sets of rankings in which we rank the algorithms from best to worst and make an inference on the model parameters and another in which we rank the algorithms from worst to best. We may expect that the results of the inferences are the same, however, this is not necessarily the case for the PL model which does not have the reversibility property.

L-Decomposability: When conducting an algorithm ranking comparison, the L-decomposability property states that the probability of an algorithm outperforming others does not depend on any other algorithm that appears in a previous position in the ranking. In our context, this is a convenient property for the considered probabilistic models on permutation spaces.

Unimodality, Consensus, and Testing: Strong unimodality is an assumption in which there is a ranking π_0 with maximum probability and given an arbitrary permutation π the probability $\Pr[\pi]$ is nonincreasing as π moves away from π_0. The permutation π moves away from π_0 when two adjacent items in π are in the same relative ordering according to π_0 and we exchange them. Complete consensus is an even more restrictive assumption that implies strong unimodality.

Given a sample of permutations, verifying that its distribution is strongly unimodal is a core issue to correctly making use of strongly unimodal probabilistic models on permutation spaces. This problem can be considered in the framework of identity testing, in which we are interested in answering a yes-or-no question about the closeness of some explicitly given distribution to an unknown distribution from which random samples are observed [21]. Despite the recent efforts to develop identity tests for ranking data, this is a challenging problem because the size of the sample space is factorial in the number of algorithms being compared [22]. In this regard, no unimodality test is known to the authors.

However, in practice, we can get some insights into these unknown distributions by observing the histogram of permutations at a given Kendall-tau distance from the ranking that appears more times in the sample. In this regard, we can expect that some models, such as the MM, are more sensitive to deviations from strong unimodality. This is because in the MM the probability of a permutation decays exponentially as its distance from the mode increases, which is an additional and more restricted assumption.

IV. Cases of Study

In this section, we compare the performance of several algorithms in two situations: 1) using synthetic data and 2) using data from real experiments. The first analysis, conducted in a controlled scenario, studies the use of the Bayesian performance analysis when the scores of the different algorithms are synthetically generated from a known probability distribution. The second situation corresponds to a real problem in which several optimization algorithms are compared when solving a benchmark set of the permutation flow shop scheduling problem (PFSP).

Bayesian Inference Settings: In our analyses, we use 1000 samples of the posterior distribution using a Markov Chain Monte Carlo method from which 500 samples are discarded as burn-in samples and the other 500 are considered for our analysis. In the specific case of the MM, we use the modified Metropolis–Hastings algorithm, introduced by Vitelli *et al.* [19]. We use the following hyperparameters for the prior distribution of the models.

1) *BT:* Concentration parameters of the Dirichlet distribution $\alpha_i = 1$ for $1 \leq i \leq n$, where n is the number of algorithms being compared.
2) *PL With Dirichlet Prior (PLD):* Concentration parameters of the Dirichlet distribution $\alpha_i = 1$ for $1 \leq i \leq n$, where n is the number of algorithms being compared. The selection of the concentration parameters $\alpha_i = 1$ corresponds to the flat Dirichlet distribution, which stands for an uninformed prior on the PL parameters.
3) *PL With Gamma Prior (PLG):* $\alpha = 0.5$ and $\beta = 0.5$ where α is the shape parameter and β the dispersion parameter of a Gamma distribution. See [18] for further details on these hyperparameters.
4) *MM:* Rate parameter $\lambda = 0.1$ and truncation parameter $\beta_{max} = 1.0 \times 10^{-6}$ for the MM dispersion parameter prior distribution. See (15) and [19] for further details on these hyperparameters.

It is worth mentioning that in our case study we are choosing uninformed priors for our analysis. However, those should not be taken as a general recommendation to be used in all scenarios. In practice, there is a myriad of recommendations in the literature to follow when selecting priors and hyperparameters and some of them have been reviewed and referenced in Section III-B.

Code and Data Availability: We provide a publicly available Python package to carry out the Bayesian analysis introduced in our work. In addition, we provide the data and presentation code used to carry out the analyses conducted in this section.[2]

1) *Python Package and Documentation:* https://pypi.org/project/BayesPermus.
2) *Data and Presentation Code:* https://github.com/ml-opt/BayesPermusPresentation.

A. Synthetically Generated Scores

In this section, we use the proposed Bayesian inference approach in the comparison of several algorithms on different problem instances using synthetically generated scores. This kind of study allows us to investigate how the Bayesian inference of the different probabilistic models on permutation spaces is affected under different circumstances. In order to obtain the synthetic dataset, we compare n independent random variables Gaussian distributed but with different location and dispersion, i.e., $X_1, \ldots, X_n$ such as $X_i \sim \mathcal{N}(\mu_i, \sigma_i)$.

Obtaining the Permutations: Given the mean and standard deviation of n Gaussian distributed random variables and a number of permutations, denoted as p, we proceed as follows.

1) Sample from each Gaussian distributed random variable, such as $x_i \sim \mathcal{N}(\mu_i, \sigma_i)$ for $1 \leq i \leq n$. As before, n is the number of algorithms being compared.

[2]The precise code to reproduce our analysis can be found in a Zenodo permanent repository at https://zenodo.org/record/6320599.

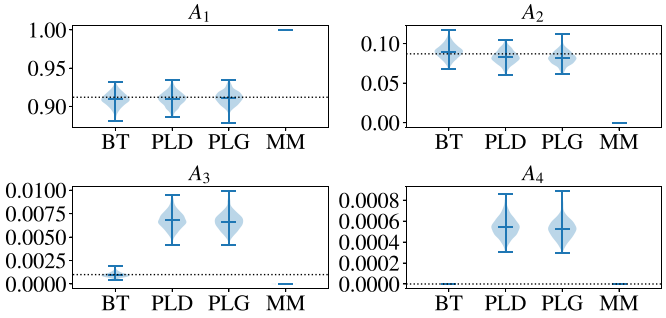

Fig. 1. Probability of each algorithm being the top-ranked algorithm using synthetically generated data. Each plot represents a different algorithm while the values represented in the horizontal axis show the results for the different Bayesian inference models.

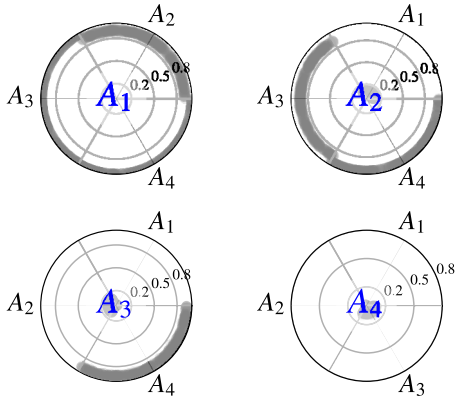

Fig. 2. Probability of each algorithm outperforming others according to the BT model. Each plot represents the probability that the algorithm in the center outperforms the others in the outer ring of the polar coordinate system.

2) Create a permutation $\pi \in S_n$ from the scores provided by each random variable, such that $x_{\pi(1)} < \cdots < x_{\pi(n)}$.
3) Repeat the previous steps p times.

Comparing the Bayesian Inferences Results: In practice, we sample 10 000 permutations from the synthetic scores and assume it is our population. This way, we can obtain the different marginal probabilities of interest from the population and take it as ground truth. Afterward, we sample a smaller number of permutations from this assumed population and conduct the Bayesian inferences on this sample. This way, we can evaluate how the different Bayesian inferences of the posterior summaries of interest deviate from the ground truth.

1) Comparing the Algorithms: We compare a fixed number algorithms $n = 4$ on several problem instances $p = 1000$. The scores of each algorithm are generated from a Gaussian distribution with mean $\mu_1 = 2.0, \mu_2 = 4.0, \mu_3 = 6.0$, and $\mu_4 = 8.0$ and standard deviation $\sigma_i = 1.0$ for $1 \leq i \leq 4$.

Probability of an Algorithm Being in the First Position: In Fig. 1, we represent the probability of each algorithm being the top-ranked algorithm (i.e., being the best performing algorithm). The figure shows violin plots [23], one per algorithm, representing the distribution of the studied probability when considering the samples of the posterior distribution $\Pr[\boldsymbol{\theta}|S]$. The black dashed horizontal line represents the empirical probability of each algorithm to be ranked the first which is obtained based on the number of times each algorithm appears in the first position of the permutations in the population.

Fig. 1 shows that the probability of being ranked first is higher for algorithm A_1 than for the other algorithms. The Bayesian inference using the PL model with the two specified priors and the BT model is slightly more in agreement with the ground truth than the inference conducted with the MM. The reference to the uncertainty of the Bayesian inference can be obtained from the variance of the posterior distributions.

Probability of an Algorithm Outperforming Others: Fig. 2 shows the probability of each algorithm, represented in the center of the plots, outperforming the other algorithms represented in the outer ring of the polar coordinate system.

Fig. 2 shows the results for just the BT model. Each plot is divided into several sectors, one sector for each of the other algorithms being compared. For example, when we are comparing A_1 with A_2, A_3, and A_4 (the top-left plot), the polar coordinate system is divided into three sectors, in which, from 0 to 360/3 degrees, we represent the probability that A_1 is better than A_2, and so on. When comparing A_1 with A_2, the plot is created as follows.

1) Draw a sample from the posterior distribution of the model parameters $\boldsymbol{\theta} \sim \Pr[\boldsymbol{\theta}|S]$.
2) Get the probability of algorithm A_1 being better than algorithm A_2, i.e., $p = \Pr[\pi^{-1}(1) < \pi^{-1}(2)|\boldsymbol{\theta}]$.
3) Create a point in the polar coordinate system (p, r) with $r \sim \text{unif}(0, 360/3)$.

We repeat this process for each posterior sample obtaining the presented plot in Fig. 2. Here, the dispersion of the distance of the dots from the origin of the polar coordinate system provides insights into the uncertainty of the estimations.

Probability of an Algorithm Being the Top-k Ranking: Fig. 3 shows the probability of each algorithm being in the top-k ranking. The black dots represent the empirical probability obtained from the population. This empirical probability is computed based on the number of times each algorithm appears in the top-k rankings of the permutations generated from the scores. This posterior summary may be of interest when we want to evaluate whether two disjoint subsets of algorithms form two groups and one of them outperforms the other. In this example, we may argue that the algorithms A_1 and A_2 outperform A_3 and A_4.

In general, we observe small discrepancies between the MM and the other models. We further investigate the causes behind such differences between the empirical distributions and the Bayesian inferences. Moreover, we evaluate whether this affects only the MM or all the studied models.

2) Multimodal Empirical Distributions: The PL, BT, and MM (with Kendall's-tau distance) share the same properties of label invariance, strong unimodality, and complete consensus. However, the MM makes additional assumptions regarding the mode and how the probability decays for permutations that move away from such mode (i.e., exponentially). We further investigate how the strong unimodality assumption made by the studied probability models is a sensible property to perform the Bayesian inference. We focus our attention on cases where the empirical distribution of the permutations does not conform to a unimodal distribution. Beyond that, we are also interested in whether the different probabilistic models studied

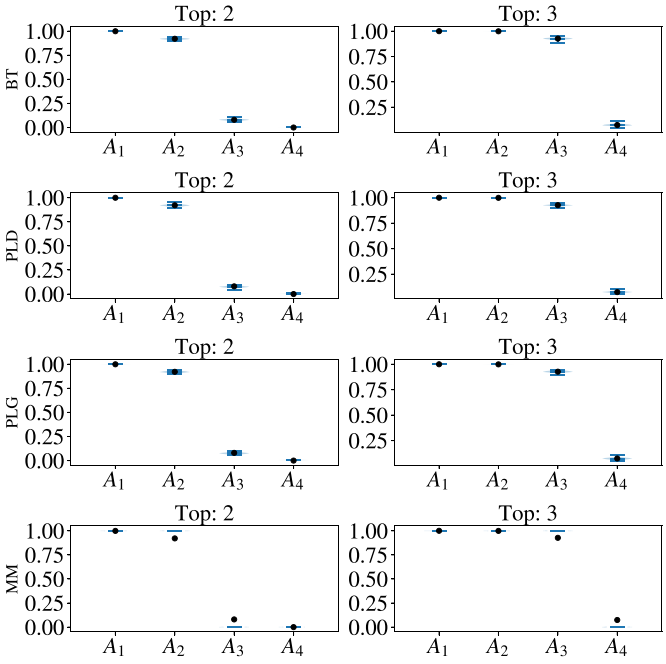

Fig. 3. Probability of each algorithm being in the top-2 and top-3 ranking. The values in the horizontal axis within each plot represent the marginal probability for the different algorithms.

in this work exhibit some differences when the unimodality assumption is not ensured in the population.

Here, we use the same data generation approach used in the previous section obtaining a number of $p = 1000$ permutations. The mean values for the Gaussian distribution of the $n = 4$ random variables used to obtain the scores of the algorithms are kept as before while we define three different configurations for the standard deviation as follows.

1) *Configuration 1:* $\sigma_1 = 2.0, \sigma_2 = \sigma_3 = 1.0$, and $\sigma_4 = 2.0$.
2) *Configuration 2:* $\sigma_1 = 4.0, \sigma_2 = \sigma_3 = 1.0$, and $\sigma_4 = 4.0$.
3) *Configuration 3:* $\sigma_1 = 12.0, \sigma_2 = \sigma_3 = 1.0$, and $\sigma_4 = 12.0$.

The idea behind the different configurations is to modify the standard deviation of the Gaussian distributed random variables which are responsible for generating the scores of algorithms A_1 and A_4. Here, as we increase the standard deviation, the probability of obtaining rankings in which A_4 outperforms A_1 increases, and therefore we can expect a multimodal empirical distribution of the rankings. Fig. 4 shows several histograms with the number of permutations at different Kendall's-tau distances from the mode. In this case, the mode is assumed to be the one with the highest frequency in the sample. Though this is not a proper identity test for unimodality, we observe that for the first configuration, the empirical distribution appears to be unimodal. Conversely, for the third configuration, the distribution seems less unimodal.

Fig. 5 shows the probability of algorithm A_1 being the best algorithm when considering the different configurations. Each column in the figure represents a different configuration,

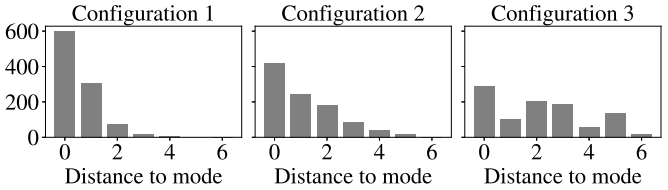

Fig. 4. Histograms showing the number of permutations in the sample at different Kendall's-tau distances from the mode.

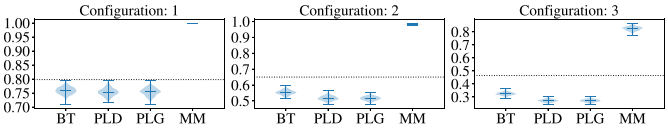

Fig. 5. Probability of algorithm A_1 being the top-ranked algorithm using different configurations of synthetically generated data.

allowing us to observe how the multimodality of the empirical distribution affects the marginal probability estimation.

In Fig. 5, as the distribution of the permutations departs from unimodality, the results are affected when compared to the ground truth. We notice that all probabilistic models are affected, but the MM seems to be more sensitive than the PL and BT models. This may be related to the previously discussed additional assumptions made by the MM. We do not provide the marginals related to the probability of an algorithm outperforming another or being in the top-k ranking in the main body of this article. However, we provide tables in our presentation code in which, in general, we observe the same trend, that is, all models are affected by the deviations from unimodality, but the MM seems more sensitive.

The unimodality requirement of the empirical distribution of the permutations is an important assumption when using the Bayesian performance analysis approach proposed in this work. We may argue that, within our specific application, unimodality is a reasonable assumption for most of the cases, as one can expect that the performance of the algorithms is consistent for all problem instances. However, this is not necessarily true for all scenarios, for example, some algorithms may perform similarly in some problem instances while in other problem instances the performance may be different.

In the next section, we develop a case study using the scores of a real comparison of optimization algorithms on several problem instances of the PFSP. The interest in the case study is based on several challenges that are not considered when using the synthetic data of the previous section: ties, repetitions, and deviations from unimodality. We use the results obtained by Ceberio *et al.* [24] in which there are several small problem instances in which most of the algorithms perform similarly (we can even find ties), while, in other larger problem instances, some algorithms are better than others. These scenarios are repeatedly reported in evolutionary optimization papers.

B. Permutation Flow Shop Scheduling Problem

Since Johnson published his work on the two-machine flow shop in 1954 [25], numerous papers have dealt with the PFSP [26]. In this section, we compare the performance of

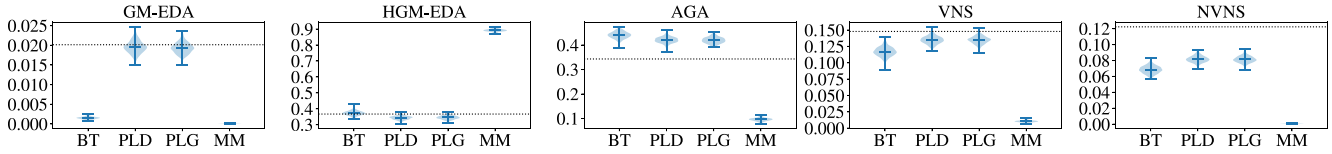

Fig. 6. Probability of each algorithm being the top-ranked algorithm using the PFSP benchmark. The values represented in the black dashed horizontal line is the ground truth obtained from all the permutations in the population of reference.

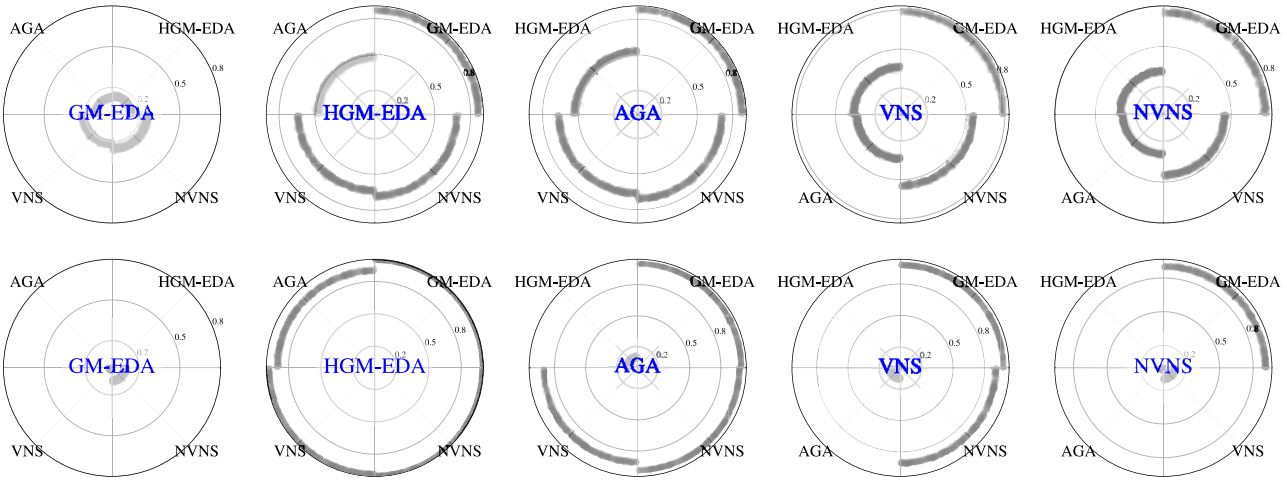

Fig. 7. Probability of each algorithm outperforming others. In the first row, we show the results obtained with the BT model and in the second with the MM. Each plot represents the probability that the algorithm in the center outperforms the other algorithms in the outer ring of the polar coordinate system.

five state-of-the-art optimization algorithms while solving a benchmarking test suite of the PFSP instances [27].

Algorithms in the Comparison: Following the work by Ceberio *et al.* [24], we compare the following algorithms.

1) *GM-EDA:* Generalized mallows EDA [24].
2) *HGM-EDA:* Hybrid GM-EDA [24].
3) *AGA:* Asynchronous genetic algorithm [28].
4) *VNS:* Variable neighborhood search [29].
5) *NVNS:* New VNS [29].

Obtaining the Permutations: We combine the instances of Taillard's benchmark test suite with the random instances generated by Ceberio *et al.* [24]. In total, there are 120 problem instances from Taillard's benchmark augmented with 220 random instances on which five algorithms are evaluated 20 times. The problem instances correspond to different configurations from 20 to 500 jobs considering 10 and 20 machines. We obtain $340 \times 20 = 6800$ permutations, such as $\pi_i \in S_5$.

Comparing the Results of the Bayesian Inference: In this case study, we assume that the permutations obtained from the scores of the algorithms define our population of reference out of which we obtain the ground truth of the different marginals of interest. From this population, we obtain a sample in which we select $p = 1000$ permutations to conduct the Bayesian analysis. With the population of reference, we can accomplish two goals: 1) to assess the results of the Bayesian analysis by comparing the posterior summaries with the ground truth and 2) to provide a sensitivity analysis on how the number of samples we take while recording the algorithm's scores affects the uncertainty of the Bayesian analysis.

Probability of an Algorithm Being Ranked the First: Fig. 6 compares the probability of the algorithms being ranked the first. The figure shows the empirical probability of each algorithm being ranked first in the black dashed horizontal line. This empirical probability is computed based on the number of times each algorithm appears in the first position of the permutations in the population of reference.

Probability of an Algorithm Outperforming Others: Fig. 7 shows the probability of each algorithm, represented in the center of the polar coordinate system, outperforming others represented in the outer ring. The first row of the plot shows the results obtained with the BT model and the second row represents the results obtained with the MM.

In general, we observe that the proposed Bayesian analysis results are in agreement with the conclusions provided in [24]. We observe that the Bayesian inferences of the different marginal probabilities are able to estimate the empirical distributions in agreement with the already available statistical conclusions of the original comparative study. In addition, we see a clear reference to the uncertainty of the estimations reflected in the variance of the posterior summaries of interest providing additional information not previously available.

Among the different probabilistic models, the MM seems to be more sensitive to the multimodal nature of the empirical distributions. See, for example, the probability of the algorithm HGM-EDA being the best algorithm according to the MM in Fig. 6. Despite the strong unimodality assumption made by the different models, in practical settings, this is not necessarily an issue for models, such as the PL and the BT.

Sensitivity to the Number of Permutations: Fig. 8 shows the probability of HGM-EDA to be the top-ranked algorithm as we take different numbers of permutations to perform the Bayesian inferences, i.e., $p = \{10, 50, 100, 200\}$. We observe

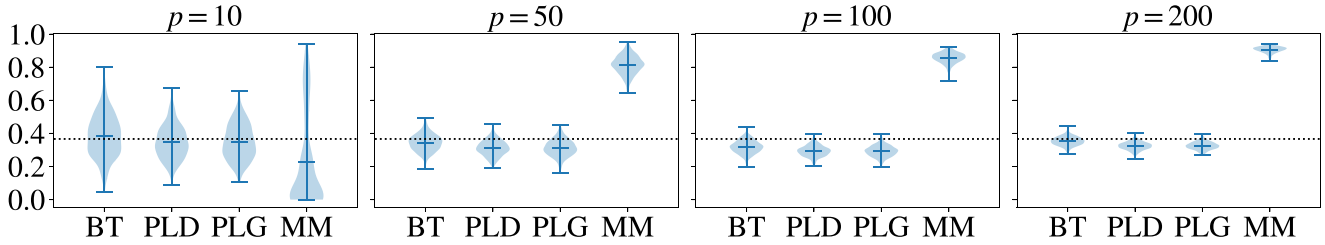

Fig. 8. Probability of algorithm HGM-EDA to be the top-ranked algorithm using $p = \{10, 50, 100, 200\}$.

that even when we take a small number of permutations, which corresponds to a situation in which we recorded the scores of the algorithms with a relatively small number of problem instances/repetitions, the Bayesian inference results are near the ground truth. As in the previous figures, the ground truth, which is represented in black dashed horizontal lines, is obtained from the permutations in the population of reference and not just the ones used for Bayesian inferences. In this case, we obtain such ground truth based on the number of times each algorithm appears in the first position of the permutations. As we increase the number of permutations in the analysis, we can observe a reduction in the variance of the posterior distribution and the uncertainty of our estimations.

V. Discussion and Outlook

Using probabilistic models in permutation spaces may be of interest to answer many questions when comparing the performance of several algorithms. In this work, we studied the use of the Bayesian inference of probabilistic models in permutation spaces to compare the performance of algorithms.

The Bayesian inference of probabilistic models in permutation spaces is a tool that allows the practitioner to quantify the uncertainty involved in the assessment of the performance of several algorithms. However, special attention requires some of the properties of the data under study, the assumptions made by the different probabilistic models in permutation spaces and the results provided by the Bayesian inference itself. In this regard, in the specific case of the BT, PL and especially for the MM, the strong unimodality assumption is an important issue to be considered by the practitioner. That is why assessing whether the empirical distribution of the permutations derived from the comparison data holds this property should be an important step before proceeding to perform the Bayesian inference with these models. We make this observation in the same fashion that some parametric tests require to verify that the data follow a Gaussian distribution. In this direction, specific tests to verify whether an empirical distribution on permutation spaces is unimodal are desired though none is known to the current authors. Another important aspect to be considered is that the marginal probability estimations provided by the studied Bayesian inference framework should not be taken as sufficient evidence that some algorithms perform better than others without carefully considering the application area.

We focused our attention on a few marginals of interest that can be obtained for the different probabilistic models. When the number of algorithms is small, the naive computation of such marginal probabilities should not be an issue, however, when this is not the case, we may face a high computational complexity. In such scenarios, closed-form expressions for the different marginal probabilities may be of interest.

In future works, exploring different probabilistic models for the Bayesian performance analysis is motivated by the fact that different models make different assumptions about the distribution of the data. In addition, while some closed-form expressions of the different marginals may not be available for some models, for others we may be able to obtain such marginal probabilities easily. Future areas of research may focus on how to visualize the marginal probabilities and explore applications in which some of them provide better insights into the performance of the different algorithms.

References

[1] M. López-Ibáñez, J. Branke, and L. Paquete, "Reproducibility in evolutionary computation," *ACM Trans. Evol. Learn. Optim.*, vol. 1, no. 4, pp. 1–21, Oct. 2021. [Online]. Available: https://doi.org/10.1145/3466624

[2] A. Benavoli, G. Corani, J. Demšar, and M. Zaffalon, "Time for a change: A tutorial for comparing multiple classifiers through Bayesian analysis," *J. Mach. Learn. Res.*, vol. 18, no. 77, pp. 1–36, 2017. [Online]. Available: http://jmlr.org/papers/v18/16–305.html

[3] R. L. Wasserstein and N. A. Lazar, "The ASA's statement on p-values: Context, process, and purpose," *Amer. Statist.*, vol. 70, no. 2, pp. 129–133, 2016.

[4] S. Goodman, "A dirty dozen: Twelve p-value misconceptions," *Seminars Hematol.*, vol. 45, no. 3, pp. 135–140, 2008.

[5] S. Greenland *et al.*, "Statistical tests, p values, confidence intervals, and power: A guide to misinterpretations," *Eur. J. Epidemiol.*, vol. 31, no. 4, pp. 337–350, 2016.

[6] J. Carrasco, S. García, M. Rueda, S. Das, and F. Herrera, "Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review," *Swarm Evol. Comput.*, vol. 54, May 2020, Art. no. 100665. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210650219302639

[7] T. Eftimov and P. Korošec, "Deep statistical comparison for multi-objective stochastic optimization algorithms," *Swarm Evol. Comput.*, vol. 61, Mar. 2021, Art. no. 100837. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210650220304909

[8] B. Calvo *et al.*, "Bayesian performance analysis for black-box optimization benchmarking," in *Proc. Genet. Evol. Comput. Conf. Companion*, New York, NY, USA, 2019, pp. 1789–1797. [Online]. Available: https://doi.org/10.1145/3319619.3326888

[9] D. I. Mattos, J. Bosch, and H. H. Olsson, "Statistical models for the analysis of optimization algorithms with benchmark functions," *IEEE Trans. Evol. Comput.*, vol. 25, no. 6, pp. 1163–1177, Dec. 2021.

[10] D. E. Critchlow, M. A. Fligner, and J. S. Verducci, "Probability models on rankings," *J. Math. Psychol.*, vol. 35, no. 3, pp. 294–318, 1991. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0022249691900504

[11] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. The method of paired comparisons," *Biometrika*, vol. 39, nos. 3–4, pp. 324–345, 1952. [Online]. Available: http://www.jstor.org/stable/2334029

[12] E. Irurozki, B. Calvo, and J. A. Lozano, "Sampling and learning mallows and generalized mallows models under the Cayley distance," *Methodol. Comput. Appl. Probabil.*, vol. 20, no. 1, pp. 1–35, 2018.
[13] F. Ghaderinezhad and C. Ley, "On the impact of the choice of the prior in Bayesian statistics," in *Bayesian Inference on Complicated Data*. London, U.K.: IntechOpen, 2019, p. 22.
[14] P. Diaconis and D. Freedman, "On inconsistent Bayes estimates of location," *Ann. Statist.*, vol. 14, no. 1, pp. 68–87, 1986. [Online]. Available: http://www.jstor.org/stable/2241268
[15] C. Ley, G. Reinert, and Y. Swan, "Distances between nested densities and a measure of the impact of the prior in Bayesian statistics," *Ann. Appl. Probabil.*, vol. 27, no. 1, pp. 216–241, 2017. [Online]. Available: https://doi.org/10.1214/16-AAP1202
[16] F. Ghaderinezhad and C. Ley, "Quantification of the impact of priors in Bayesian statistics via Stein's method," *Statist. Probabil. Lett.*, vol. 146, pp. 206–212, Mar. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167715218303596
[17] J. I. Yellott, "The relationship between Luce's choice axiom, Thurstone's theory of comparative judgment, and the double exponential distribution," *J. Math. Psychol.*, vol. 15, no. 2, pp. 109–144, 1977. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0022249677900268
[18] J. Guiver and E. Snelson, "Bayesian inference for Plackett–Luce ranking models," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, New York, NY, USA, 2009, pp. 377–384. [Online]. Available: https://doi.org/10.1145/1553374.1553423
[19] V. Vitelli, Ø. Sørensen, M. Crispino, A. Frigessi, and E. Arjas, "Probabilistic preference learning with the mallows rank model," *J. Mach. Learn. Res.*, vol. 18, no. 158, pp. 1–49, 2018. [Online]. Available: http://jmlr.org/papers/v18/15–481.html
[20] F. Collas and E. Irurozki, "Concentric mixtures of mallows models for top-*k* rankings: Sampling and identifiability," in *Proc. 38th Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 2079–2088. [Online]. Available: http://proceedings.mlr.press/v139/collas21a.html
[21] R. I. Busa-Fekete, D. Fotakis, and M. Zampetakis, "Private and non-private uniformity testing for ranking data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 9480–9492. [Online]. Available: https://openreview.net/forum?id=lMrwT4C93eT
[22] R. I. Busa-Fekete, D. Fotakis, B. Szorenyi, and M. Zampetakis, "Identity testing for mallows model," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 23179–23190. [Online]. Available: https://openreview.net/forum?id=M7emZFOLbH
[23] J. L. Hintze and R. D. Nelson, "Violin plots: A box plot-density trace synergism," *Amer. Statist.*, vol. 52, no. 2, pp. 181–184, 1998. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/00031305.1998.10480559
[24] J. Ceberio, E. Irurozki, A. Mendiburu, and J. A. Lozano, "A distance-based ranking model estimation of distribution algorithm for the Flowshop scheduling problem," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 286–300, Apr. 2014.
[25] S. M. Johnson, "Optimal two- and three-stage production schedules with setup times included," *Nav. Res. Logist. Quart.*, vol. 1, no. 1, pp. 61–68, 1954. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800010110
[26] J. N. Gupta and E. F. Stafford, "Flowshop scheduling research after five decades," *Eur. J. Oper. Res.*, vol. 169, no. 3, pp. 699–711, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221705001372
[27] E. Taillard, "Benchmarks for basic scheduling problems," *Eur. J. Oper. Res.*, vol. 64, no. 2, pp. 278–285, 1993. [Online]. Available: https://www.sciencedirect.com/science/article/pii/037722179390182M
[28] X. Xu, Z. Xu, and X. Gu, "An asynchronous genetic local search algorithm for the permutation flowshop scheduling problem with total flowtime minimization," *Expert Syst. Appl.*, vol. 38, no. 7, pp. 7970–7979, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417410014387
[29] W. E. Costa, M. C. Goldbarg, and E. G. Goldbarg, "New VNS heuristic for total flowtime flowshop scheduling problem," *Expert Syst. Appl.*, vol. 39, no. 9, pp. 8149–8161, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417412001728

Jairo Rojas-Delgado received the bachelor's degree in software engineering, the master's degree in 2018, and the Ph.D. degree in artificial intelligence and machine learning in 2020 from the University of Informatics Sciences, Havana, Cuba.

He is a Postdoctoral Fellow with the Basque Center for Applied Mathematics, Bilbao, Spain. He is interested in optimization, graph neural networks, and high-performance computing.

Josu Ceberio received the Ph.D. degree from the University of the Basque Country (UPV/EHU), Leioa, Spain, in 2014.

He is currently an Associate Professor with the Department of Computer Science and Artificial Intelligence, University of the Basque Country (UPV/EHU). His main research areas are evolutionary computation, combinatorial optimization problems, and reinforcement learning.

Dr. Ceberio has been a member of the Editorial Board of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION since 2022.

Borja Calvo received the bachelor's degree in computer science, the master's degree in biochemistry, and the Ph.D. degree in computer science from the University of the Basque Country, Leioa, Spain, in 1999, 2004, and 2008, respectively.

After two years as a Postdoctoral Researcher with the Intelligent Systems Group, in 2011, he won his current Lecturer position with the Department of Computer Science and Artificial Intelligence, University of the Basque Country. He is also supervising three Ph.D. students and several master's thesis.

Jose A. Lozano (Fellow, IEEE) received the Ph.D. degree in computer science from the University of the Basque Country, Leioa, Spain, in 1998.

He is currently a Full Professor with the University of the Basque Country and the Scientific Director of Basque Center for Applied Mathematics, Bilbao, Spain. His major research interests include machine learning and evolutionary computation and their application in biomedicine.

Prof. Lozano has been an Associate Editor of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.