Technical Report

UNIVERSITY OF THE BASQUE COUNTRY
Department of Computer Science and Artificial Intelligence

Extending Distance-based Ranking Models In Estimation of Distribution Algorithms

Josu Ceberio, Ekhine Irurozki,
Alexander Mendiburu, Jose A. Lozano

May, 2014

San Sebastian, Spain
www.ehu.es/ccia-kzaa
hdl.handle.net/10810/4562

Extending Distance-based Ranking Models in Estimation of Distribution Algorithms

Josu Ceberio, Ekhine Irurozki, Alexander Mendiburu
and Jose A. Lozano (*Member, IEEE*)

***Abstract*— Recently, probability models on rankings have been proposed in the field of estimation of distribution algorithms in order to solve permutation-based combinatorial optimisation problems. Particularly, distance-based ranking models, such as Mallows and Generalized Mallows under the Kendall's-τ distance, have demonstrated their validity when solving this type of problems. Nevertheless, there are still many trends that deserve further study. In this paper, we extend the use of distance-based ranking models in the framework of EDAs by introducing new distance metrics such as Cayley and Ulam. In order to analyse the performance of the Mallows and Generalized Mallows EDAs under the Kendall, Cayley and Ulam distances, we run them on a benchmark of 120 instances from four well known permutation problems.**

The conducted experiments showed that there is not just one metric that performs the best in all the problems. However, the statistical test pointed out that Mallows-Ulam EDA is the most stable algorithm among the studied proposals.

I. Introduction

In combinatorics, many optimisation problems are defined as "the way of arranging n number of objects" such that a specific criterion is maximised (or minimised). Codified naturally as permutations, these problems, referred to as *permutation-based problems*, are a subset of NP-Complete combinatorial optimisation problems. Vehicle routing [1], job scheduling [2] or assignment problems [3] are some of the several examples that can be found in the literature. Due to their high complexity and relevance, permutation-based problems have been frequently addressed in the field of combinatorial optimization.

Among the wide variety of exact, heuristic and meta-heuristic algorithms, Branch and Bound [4], Constructive Heuristics [5], Local Search [6], Genetic Algorithms [7], Ant Colony Optimization [8], Particle Swarm Optimization [9], or Estimation of Distribution Algorithms (EDA) [10], [11] are a few of the algorithms that have been proposed in the combinatorial optimisation literature.

Josu Ceberio, Ekhine Irurozki and Jose A. Lozano are with the Department of Computer Science and Artificial Intelligence, University of the Basque Country UPV/EHU, Donostia, Spain (email: {josu.ceberio, ekhine.irurozqui, ja.lozano}@ehu.es).

Alexander Mendiburu is with the Department of Computer Architecture and Technology, University of the Basque Country UPV/EHU, Donostia, Spain (email: alexander.mendiburu@ehu.es).

This work has been partially supported by the Saiotek and Research Groups 2013-2018 (IT-609-13) programs (Basque Government), TIN2010-14931 (Ministry of Science and Technology), COMBIOMED network in computational bio-medicine (Carlos III Health Institute), and by the NICaiA Project PIRSES-GA-2009-247619 (European Commission). Josu Ceberio holds a grant from Basque Government.

In this paper, we are particularly interested in the development of EDAs [12], [13] for solving permutation-based problems. EDAs are a type of evolutionary algorithm that, based on machine learning techniques, learn at each iteration a probabilistic model from a set of candidate solutions in order to capture the most relevant information. By sampling the model, EDAs guide the search towards promising areas of the search space. Numerous papers have demonstrated the validity of EDAs when solving combinatorial optimisation problems [11], [13]–[18]. Permutation-based problems, however, present a real challenge for EDAs, since in most cases, the typical compact and factorized probability models cannot capture the *mutual exclusivity constraints* associated with permutations [19].

Recently, a number of papers have proposed using probability models on rankings in the framework of EDAs [20]–[23]. Ceberio et al. [20] published the first attempt of using probability models on rankings in the framework of EDAs. In that work, a *distance-based ranking model* called Mallows model (MM) [24]–[27] was used. This model, defined by two parameters, a central permutation σ_0 and a spread parameter θ, is analogous to the Gaussian distribution over the domain of permutations. As an extension to the MM, the Generalized Mallows EDA was presented in [21]. Proposed for the first time by Fligner et al. [28], the Generalized Mallows model (GMM) is defined by a central permutation σ_0 and a vector of $n-1$ spread parameters $\boldsymbol{\theta}$, each of which affect a particular component of the solution.

The MM and the GMM assign to each permutation in the search space a probability that decays exponentially with respect to its distance to σ_0. Commonly, the Kendall's-τ metric is the distance used to learn and sample these models [20], [21]. Nevertheless, there exist other distance metrics that could be studied beyond the Kendall's-τ [27]. In this sense, with the aim of exploring other possibilities, in this paper we extend previous works by introducing efficient implementations of the Mallows and Generalized Mallows models for the Cayley [29] and Ulam [30] distances.

In order to study the performance of the Kendall's-τ, Cayley and Ulam metrics in the Mallows and the Generalized Mallows EDAs, we test these algorithms on a benchmark of 120 instances from the Traveling Salesman Problem (TSP), the Quadratic Assignment Problem (QAP), the Linear Ordering Problem (LOP) and the Permutation Flowshop Scheduling Problem (PFSP). The conducted experiments show that there is not just one algorithm that performs the best in all the proposed instances. However, the statistical analysis

concluded that the Mallows EDA under Ulam distance is the preferred algorithm due to its stable performance in all the problems. In addition, the experiments reveal that Mallows EDA under Cayley is the algorithm that performs the worst.

The remainder of the paper is organised as follows: in the next section, the permutation problems considered in the experimental study are briefly introduced. Afterwards, in Section III the Mallows and the Generalized Mallows models are described in detail. In Section IV, Kendall's-τ, Cayley and Ulam distances are introduced, and their respective learning and sampling procedures are detailed. In Section V, an experimental study of the Mallows and the Generalized Mallows EDAs under the different distance metrics is performed. Finally, some conclusions and ideas for future work are presented in Section VI.

II. Permutation-based Combinatorial Optimization Problems

Permutation-based problems are combinatorial optimization problems whose solutions can be naturally represented as a permutation. A permutation is understood as a bijection σ of indexes $\{1,\dots,n\}$ onto $\{1,\dots,n\}$, where $\sigma(i)$ (also denoted as σ_i)[1] denotes the item at position i, and $\sigma^{-1}(i)$ stands for the position of item i in σ (denoted also as $\sigma\langle i\rangle$).

In what follows, we briefly describe the problems we consider in this paper.

A. Linear Ordering Problem

Given a matrix $B = [b_{ij}]_{n\times n}$ of numerical entries, the Linear Ordering Problem (LOP) [31], [32] consists of finding a simultaneous permutation σ of the rows and columns of B such that the sum of the entries above the main diagonal is maximised (or equivalently, the sum of the entries below the main diagonal is minimised). The equation below formalises the LOP function:

$$f(\sigma) = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} b_{\sigma_i\sigma_j}$$

where σ_i denotes the index of the row (and column) located at position i in the solution σ. A particular feature of this problem worth noting is that the contribution of an index σ_i to the objective function depends on the previous and posterior sets of indexes, but not on their relative ordering [32].

B. Permutation Flowshop Scheduling Problem

In the permutation flowshop scheduling problem (PFSP) [33], n jobs $(i = 1,\dots,n)$ have to be scheduled on m machines $(j = 1,\dots,m)$ in such a way that a given criterion is minimized. A job consists of m operations and the j-th operation of each job must be processed on machine j for a given specific processing time without interruption. The processing times are fixed, non-negative values and every job is available at time zero. At a given time, a job can start on the j-th machine when its $(j-1)$-th operation has finished on machine $(j-1)$, and machine j is free. A solution for the problem is codified as a permutation σ of length n where σ_i denotes the job at position i.

With respect to the optimisation criterion, we considered the total flow time (TFT), which optimises the sum of the completion times of each job. Eq. 1 expresses mathematically the concept of TFT for a solution σ, where $c_{\sigma_i,m}$ stands for the completion time of job σ_i on machine m.

$$F(\sigma) = \sum_{i=1}^{n} c_{\sigma_i,m} \tag{1}$$

Being $p_{\sigma_i,j}$ the processing time required by job σ_i on machine j, the completion time of job σ_i on machine j can be recursively calculated as:

$$c_{\sigma_i,j} = \begin{cases} p_{\sigma_i,j} & i = j = 1 \\ p_{\sigma_i,j} + c_{\sigma_{i-1},j} & i > 1, j = 1 \\ p_{\sigma_i,j} + c_{\sigma_i,j-1} & i = 1, j > 1 \\ p_{\sigma_i,j} + \max\{c_{\sigma_{i-1},j}, c_{\sigma_i,j-1}\} & i > 1, j > 1 \end{cases}$$

Note that the completion time of each job σ_i depends on the ordering of the previous $\{\sigma_1,\dots,\sigma_{i-1}\}$ jobs, and therefore the contribution of each job is highly determined by its processing times as well as by the ordering of the previously scheduled jobs.

C. Traveling Salesman Problem

The Traveling Salesman Problem (TSP) [34] consists of looking for the shortest path, in terms of time, distance, or any similar criterion, to go over n different cities visiting each city only once and returning to the city of departure. A solution for the problem is codified as a permutation σ of cities where σ_i denotes the city visited in the i-th position. The objective function f is defined as the sum of the distances of going from city $i-1$ to i, denoted as d_{ij}, through all cities in the order specified in σ:

$$f(\sigma) = \sum_{i=2}^{n} d_{\sigma_{i-1}\sigma_i} + d_{\sigma_n\sigma_1}$$

In the TSP we note that due to the cyclic nature of the solutions, the relevant information to calculate the fitness function of a solution σ is given by the relative ordering of the cities in the permutation, and not by their absolute position, which in this case is irrelevant.

D. Quadratic Assignment Problem

The Quadratic Assignment Problem (QAP) [35] is the problem of allocating a set of facilities to a set of locations, with a cost function associated to the distance and flow between the facilities. The objective is to assign each facility to a location such that the total cost is minimized. Specifically, given two $n\times n$ numerical matrices $H = [h_{ij}]$ and $D = [d_{ij}]$, where h_{ij} is the flow between facility i and facility j, and d_{ij} denotes the distance between the location i and j, the goal

[1]With readability purposes we will use $\sigma(i)$ and σ_i interchangeably throughout the paper.

is to find a permutation σ (where σ_i represents the facility allocated at position i), such that the function

$$f(\sigma) = \sum_{i=1}^{n}\sum_{j=1}^{n} h_{\sigma_i\sigma_j} * d_{ij}$$

is minimised. In this problem, the quality of the solution depends on the absolute position of each facility in the permutation.

III. Mallows and Generalized Mallows models

The Mallows model (MM) [24] is one of the most popular probability models for permutation spaces. Under this model, the probability value of every permutation $\sigma \in \mathbb{S}_n$ (where $\mathbb{S}_n$ stands for the set of $n!$ permutations of n items) depends on just two parameters: a spread parameter θ, and the distance to a central permutation σ_0, which is calculated by a particular metric $D(\sigma, \sigma_0)$. Formally, the MM is defined as follows:

$$P(\sigma) = \psi(\theta)^{-1} exp(-\theta D(\sigma, \sigma_0)) \tag{2}$$

where $\psi(\theta)$ is a normalization constant. When $\theta > 0$, the central permutation σ_0 is the mode of the distribution. The model assigns to each permutation $\sigma \in \mathbb{S}_n$ a probability that decays exponentially with respect to its distance to σ_0. On the one hand, the larger the value of θ, the more peaked the distribution becomes around the central permutation. On the other hand, when θ equals 0, Eq. 2 assigns equal probability to every permutation σ in $\mathbb{S}_n$, and for $\theta < 0$ then σ_0 is the antimode.

As an extension to the MM, the Generalized Mallows model (GMM) was proposed in [28]. Under the GMM, the central permutation σ_0 is also the mode of the distribution. However, instead of a single spread parameter θ, the GMM makes use of a vector of $n-1$ spread parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_{n-1})$, each θ_j affecting a particular position j in the permutation. This allows modelling a distribution with more emphasis on the consensus of certain positions of the permutation while having more uncertainty in some of the others. The GMM requires the metric to be decomposed into $n-1$ terms in such a way that it can be expressed as

$$D(\sigma, \sigma_0) = \sum_{j=1}^{n-1} S_j(\sigma\sigma_0^{-1}) \tag{3}$$

where σ_0^{-1} stands for the inverse permutation of σ_0, $\sigma\sigma_0^{-1}$ denotes the composition operation between σ and σ_0^{-1} and $S_j(\sigma\sigma_0^{-1})$ is the term associated to the position j. For such a metric, the GMM model is formalised as

$$P(\sigma) = \psi(\boldsymbol{\theta})^{-1} exp(\sum_{j=1}^{n-1} -\theta_j S_j(\sigma\sigma_0^{-1})) \tag{4}$$

In order to introduce these models in the framework of EDAs, it is necessary to define efficient learning and sampling methods for each model-metric. As regards the learning, this process consists of two steps that are similar for all the model-metric combinations: first, given a sample of permutations, the consensus permutation σ_0 is calculated, and then, the spread parameter θ for MM (or $\boldsymbol{\theta}$ for GMM) is estimated. Usually, this process is approached via maximum likelihood estimation (MLE). However, the time required for an exact learning scales factorially with the number of items [21], [29], [30], and thus, in this work we carry out an approximate learning of the parameters:

1) The estimated consensus permutation $\hat{\sigma}_0$ is calculated as the *set median permutation*, which is the permutation in the sample that minimizes the sum of the distances to the rest of the permutations in the sample. Although one can find in the literature many approximation algorithms for the MLE for the consensus permutation [36], we have decided to use the set median permutation in this work for two reasons: (1) Computing the set median permutation is a quick process, carried out in time $O(dm^2)$, where d denotes the time complexity of computing the distance between two permutations and m the number of permutations in the sample. (2) Using the same approximation method for the consensus permutation for every distance metric, we try to compare the EDAs independently of the quality of the learning algorithm.
2) Once $\hat{\sigma}_0$ is estimated, the MLE for the spread parameter θ for MM (or $\boldsymbol{\theta}$ for GMM) is computed. The expression for this parameter is obtained by equaling to zero the derivative of the likelihood. This expression differs depending on the distance metric, however, we solved them by means of the Newton-Raphson algorithm. For further details we refer the interested reader to [21], [37].

As regards the sampling process, each metric has a particular procedure, and thus, in the next section we introduce the Kendall's-τ, Cayley and Ulam metrics in detail, including the method for sampling (obtain a new permutation).

IV. Distance Metrics

Due to its applications in preference modelling, Kendall's-τ metric is the metric that has captured the attention of the research community the most [38]–[40]. However, recently, new techniques for efficiently learning and sampling the MM and the GMM models for Cayley [29] and Ulam [30] distance metrics have been proposed. This fact has encouraged us to study the performance of the MM and the GMM under these three distance metrics inside the framework of EDAs.

Before going into details, it is important to point out that the three metrics considered in this study are *right invariant*, which means that $D(\sigma, \pi) = D(\sigma\pi^{-1}, \pi\pi^{-1}) = D(\sigma\pi^{-1}, e)$. e stands for the identity permutation of size n, $(1, 2, \ldots, n)$, as a consequence, the distance from/to e is denoted as a one parameter function so $D(\sigma\pi^{-1}, e) = D(\sigma\pi^{-1})$.

In what follows, a detailed description of the metrics is introduced:

A. Kendall's-τ distance

The Kendall's-τ distance $D_\tau(\sigma, \pi)$ measures the number of pairs of items for which σ and π have opposing ordering,

or equivalently the minimum number of adjacent transpositions needed to bring σ^{-1} into π^{-1}. It can be computed in $O(n^2)$.

$D_\tau(\pi)$ can be decomposed in $n-1$ terms (as in Eq. 3) as $V_j(\pi) = \sum_{i=j+1}^{n} I_{[\pi(j)>\pi(i)]}$, where $I[\cdot]$ denotes the indicator function. Basically, $V_j(\pi)$ is the number of positions of the permutation on the right of j with values smaller than $\pi(j)$. It follows from the definition that $V_j(\pi)$ ranges from 0 to $n-j$ for $1 \leq j < n$. This decomposition allows the GMM given in Eq. 4 to be explicitly written for the Kendall's-τ distance as follows:

$$P(\sigma) = \psi(\boldsymbol{\theta})^{-1} exp(\sum_{j=1}^{n-1} -\theta_j V_j(\sigma\sigma_0^{-1})) \tag{5}$$

Under this distance, the normalisation constant $\psi(\boldsymbol{\theta})$, which if calculated naively requires $n!$ sums, can be simplified as the product of $n-1$ terms,

$$\psi(\boldsymbol{\theta}) = \prod_{j=1}^{n-1} \psi_j(\theta_j) = \prod_{j=1}^{n-1} \frac{1 - exp(-\theta_j(n-j+1))}{1 - exp(-\theta_j)} \tag{6}$$

1) Sampling the distribution: The moment generating function for $V(\pi) = (V_1(\pi), \ldots, V_{n-1}(\pi))$ defines the probability of each $V_j(\pi)$ as follows [26]:

$$P(V_j(\sigma\sigma_0^{-1}) = r_j) = \frac{exp(-\theta_j r_j)}{\psi_j(\theta_j)} \qquad r_j \in \{0, ..., n-j\} \tag{7}$$

Moreover, there is a bijection between each possible value of $V(\pi)$ and each possible $\sigma \in \mathbb{S}_n$. As a consequence, the sampling process for each permutation is carried out in three stages. First, from Eq. 7, we randomly generate a vector $V(\sigma\sigma_0^{-1})$. Then, from this vector we calculate the associated $\sigma\sigma_0^{-1}$. The final permutation σ is obtained by composing $\sigma\sigma_0^{-1}$ with σ_0, so $\sigma\sigma_0^{-1}\sigma_0 = \sigma e = \sigma$. The complexity of sampling σ under the Kendall's-τ distance is $O(n^2)$.

B. Cayley distance

The Cayley distance $D_c(\sigma, \pi)$ counts the minimum number of swaps (not necessary adjacent) to convert σ into π. It can be computed in $O(n)$.

This distance is related to the cyclic structure of permutations. The Cayley distance $D_c(\pi)$ can be decomposed in $n-1$ boolean terms $D_c(\pi) = \sum_{j=1}^{n-1} X_j(\pi)$ where $X_j(\pi) = 0$ iff j is the largest item of a cycle in π, and 1 otherwise. Since this decomposition complies with Eq. 3, the GMM under the Cayley distance can be given as follows:

$$P(\sigma) = \psi(\boldsymbol{\theta})^{-1} exp(\sum_{j=1}^{n-1} -\theta_j X_j(\sigma\sigma_0^{-1})) \tag{8}$$

for every $\sigma \in \mathbb{S}_n$. The normalisation constant $\psi(\boldsymbol{\theta})$, under the Cayley distance, is formalised as

$$\psi(\boldsymbol{\theta}) = \prod_{j=1}^{n-1} \psi_j(\theta_j) = \prod_{j=1}^{n-1} (n-j) exp(-\theta_j) + 1 \tag{9}$$

1) Sampling the distribution: In this case, the moment generating function is given by the probability of each $X_j(\pi)$ as follows [26]:

$$p(X_j(\sigma\sigma_0^{-1}) = 1) = \frac{(n-j) exp(-\theta_j)}{\psi_j(\theta_j)} \tag{10}$$

Unfortunately, although each π has a unique $X(\pi) = (X_1(\pi), \ldots, X_{n-1}(\pi))$, the opposite is not necessarily true. And thus, a method for the random generation of a permutation π given $X(\pi)$ [29] is used. According to that method, the process of sampling a solution from a GMM model under the Cayley distance can be carried out in three stages: First, from Eq. 10, we randomly generate a boolean vector $X(\sigma\sigma_0^{-1})$. Secondly, we calculate the associated $\sigma\sigma_0^{-1}$ with the techniques described in [29]. Finally, by right invariance we obtain the final permutation σ. The complexity of sampling σ under the Cayley distance is $O(n^2)$.

C. Ulam distance

The Ulam distance between two permutations σ and π, $D_u(\sigma, \pi)$, is exactly the size of the complement of the longest common subsequence of σ and π or, equivalently, n minus the length of the longest increasing subsequence (LIS) of $\sigma\pi^{-1}$. Therefore, $D_u(\pi)$ is n minus the length of the LIS in π. It can be exactly computed in $O(n\ log\ n)$. Since the Ulam distance can not be decomposed as in Eq. 3, the GMM can not be coupled with the Ulam distance, and thus, just the MM case is considered. It is expressed as follows:

$$P(\sigma) = \psi(\boldsymbol{\theta})^{-1} exp(-\theta D_u(\sigma, \sigma_0)) \tag{11}$$

Unfortunately, there is no closed form for $\psi(\theta)$. Both exact and approximate expressions for $\psi(\theta)$ can be found in [30]. Due to the lack of space in this paper, the algebraic machinery has not been included. However, we give a brief intuition of the process here. Let $S_u(n, d)$ be the number of permutations of n items at Ulam distance d. Then, $\psi(\theta)$ can be given as follows.

$$\psi(\theta) = \sum_{d=0}^{n-1} S_u(n, d) exp(-\theta d) \tag{12}$$

1) Sampling the distribution: Sampling the MM is based on the fact that every permutation at the same distance has equal probability. Note that the probability of obtaining a permutation at distance d from the identity permutation is as follows.

$$P(\pi | D_u(\pi) = d) = \psi(\theta)^{-1} S_u(n, d) exp(-\theta d) \tag{13}$$

In this way, the process of sampling a solution from the model can be performed in three stages. First, by means of Eq. 13, randomly select a distance d. In order to speed the process up, the approximated version of the algorithm restricts the maximum distance at which to sample to $d_{\max} = n/2$. Then, generate uniformly at random a permutation at distance d from e, using the techniques detailed in [30]. Finally, by right invariance we obtain the final permutation σ.

TABLE I: Summary of the most successful interpretations (a or b) for each algorithm-problem pair. a/b denotes that no statistical differences were found.

EDAs	LOP	PFSP	QAP	TSP
M_k	b	b	b	a/b
M_c	a/b	a/b	a/b	a/b
M_u	a/b	b	b	b
GM_k	b	b	b	a/b
GM_c	b	b	b	a/b

In the previous sections, the methods for learning and sampling the MM and the GMM under the Kendall's-τ, Cayley and Ulam distance were introduced. However, there is a final concern that needs to be addressed in order to integrate these models into the framework of EDAs. We refer to the interpretation of the individuals (permutation) when calculating their fitness in the evaluation step. Given an individual σ, there exist two possible interpretations of the solution: $a)$ to consider σ_i as the item at position i (introduced in Section II), or $b)$ to consider σ_i as the position at which item i is located. Let $\sigma = (2, 3, 1)$ be a solution for the PFSP. Interpretation $a)$ considers that job 2 is scheduled first, next job 3 and job 1 is scheduled last. Inversely, interpretation $b)$ considers that job 1 is ranked in the 2^{nd} position ($\sigma_1 = 2$), job 2 is ranked 3^{rd}, and job 3 is ranked 1^{st}.

According to our experience, we think that the appropriate interpretation of the solutions is b. Nevertheless, in order to confirm our intuition we have performed some experiments with the EDAs proposed in the following section for both interpretations. Moreover, in order to assess whether there exist statistical differences between the two interpretations, we have applied a non-parametric Wilcoxon test to the average results obtained. A level of significance $\alpha = 0.05$ was set. Table I shows a summary of the most successful interpretation in each case. In the view of the results, we will use interpretation b as the preferred one in the experimental study.

V. Experiments

In order to compare the performance of the Kendall's-τ, Cayley and Ulam distances within the Mallows and Generalized Mallows models, we ran five different EDAs: the Kendall's-τ (M_k), Cayley (M_c) and Ulam (M_u) Mallows EDAs, and the Kendall's-τ (GM_k) and Cayley (GM_c) Generalized Mallows EDAs. Recall that the GMM of the Ulam is not defined.

All the EDAs were implemented in C++ programming language. The experimentation was conducted on a cluster of 20 nodes, each of them equipped with two Intel Xeon X5650 CPUs and 48GB of memory.

In relation to the experimentation instances, a benchmark of 120 instances of the TSP, QAP, LOP and PFSP problems was proposed (30 instances of each problem). The instances of the TSP were downloaded from TSPLIB [41], and the instances of the QAP and PFSP were obtained from the Taillard's Benchmark [42]. As regards the LOP, the smallest 15 instances were obtained from the LOLIB benchmark, and the rest were artificially generated as specified in [32][2].

A. Parameter Settings

In the list below we summarise the parameters employed in the EDAs:

- Population size is set to $10n$.
- n individuals are selected to learn the model.
- $10n - 1$ individuals are sampled at each generation.
- Elitism criteria is used.
- A maximum number of $1000n^2$ evaluations are considered as stopping criterion.

Other particular settings of the algorithms:

- A maximum number of 100 iterations, and a minimum accuracy improvement of 0.001 are used in the Newton-Raphson procedure.
- For feasibility purposes, θ values range in [0,10] for the three metrics.

B. Results

Each *algorithm - instance* pair was run 10 times. The performance measure employed in our study is the average relative percentage deviation (ARPD):

$$ARPD = \frac{|AvgRes - Best|}{Best}$$

where $AvgRes$ denotes the average results obtained throughout the 10 repetitions, $Best$ stands for the best solution obtained throughout the experimental study by any of the EDAs. The ARPD results of the executions are collected in Table II. Results in bold correspond to the algorithm that obtained the lowest ARPD (best) among the compared approaches.

The conducted experiments show that the performance of the proposed EDAs vary depending on the problems. In the following list we summarise the results obtained for each problem, highlighting the most remarkable results:

- In the LOP, we observe that M_u is the algorithm that most frequently obtained the best results, in 18 instances out of 30.
- As regards the QAP benchmark, we appreciate that the size of the instance has a remarkable influence on the results. GM_c obtained the best result for 17 instances, the small ones, whereas M_u obtained the best results for 10 instances (mostly large).
- In the PFSP, M_u and GM_c were the best performing algorithms. M_u obtained the best results in 16 instances out of 30, and GM_c was the best in the remaining 14 instances.
- In the TSP, although we do not have all the results for the M_u, this algorithm is clearly the best performing EDA, obtaining the best result in 20 out of 23 completed instances. In the remainder instances, GM_c is the algorithm that stands out over the rest.

[2]Supplementary results, source codes, instances, and extended material of the experiments can be downloaded from http://www.sc.ehu.es/ccwbayes/members/jceberio/CEC2014/CEC2014.html.

TABLE II: ARPD results of 10 repetitions of the EDAs for full benchmark of instances. Results in bold denote the best performing algorithm. Missing results, denoted as '-', indicate the executions that, due to their computational cost, did not finish.

Problem	Instance	Size	M_k	M_c	M_u	GM_k	GM_c	Problem	Instance	Size	M_k	M_c	M_u	GM_k	GM_c
LOP	N-t59b11xx	44	0.0166	0.1376	0.0265	**0.0127**	0.0188	QAP	tai15a	15	0.0619	0.0248	0.0641	0.0571	**0.0213**
	N-t59d11xx	44	0.0198	0.1201	**0.0018**	0.0120	0.0109		tai15b	15	0.0079	0.0040	0.0098	0.0083	**0.0029**
	N-t59f11xx	44	0.0217	0.1336	0.0429	0.0190	**0.0167**		nug17	17	0.0362	**0.0160**	0.0628	0.0478	0.0223
	N-be75eec	50	0.0211	0.1462	**0.0030**	0.0157	0.0788		nug18	18	0.0403	0.0235	0.0686	0.0439	**0.0130**
	N-be75np	50	0.0159	0.1560	0.0518	0.0131	**0.0034**		nug20	20	0.0553	0.0456	0.0790	0.0644	**0.0285**
	N-be75oi	50	0.0072	0.0865	**0.0017**	0.0098	0.0525		tai20a	20	0.0985	0.0838	0.0814	0.1002	**0.0451**
	N-be75tot	50	0.0261	0.1721	**0.0010**	0.0200	0.0449		tai20b	20	0.0460	0.0154	0.1282	0.0653	**0.0115**
	N-tiw56r58	56	0.0357	0.1775	**0.0016**	0.0152	0.0139		nug21	21	0.1204	0.0695	0.0621	0.1082	**0.0218**
	N-tiw56r66	56	0.0360	0.1662	0.0259	**0.0185**	0.0246		tai25a	25	0.0693	0.0700	0.0578	0.0760	**0.0433**
	N-tiw56r67	56	0.0364	0.1558	0.0458	**0.0183**	0.0284		tai25b	25	0.1098	0.2101	0.1941	0.1799	**0.0132**
	N-tiw56r72	56	0.0317	0.1615	**0.0020**	0.0160	0.0332		bur26a	26	0.0173	0.0126	0.0102	0.0139	**0.0011**
	N-stabu70	60	0.0389	0.1532	**0.0015**	0.0298	0.0738		bur26b	26	0.0164	0.0090	0.0097	0.0176	**0.0012**
	N-stabu74	60	0.0397	0.1475	**0.0026**	0.0196	0.0772		bur26c	26	0.0199	0.0092	0.0135	0.0152	**0.0008**
	N-stabu75	60	0.0415	0.1443	**0.0012**	0.0269	0.0728		bur26d	26	0.0216	0.0094	0.0137	0.0207	**0.0021**
	N-usa79	79	0.0551	0.1321	**0.0385**	0.0548	0.0738		tai30a	30	0.0764	0.0742	**0.0579**	0.0778	0.0646
	N-t65d11xx	100	0.1672	0.2233	**0.1388**	0.1648	0.1748		tai30b	30	0.1213	0.1988	0.0893	0.0964	**0.0492**
	N-t65f11xx	100	0.1564	0.2028	**0.1096**	0.1515	0.1650		tai35a	35	0.0232	0.0258	**0.0073**	0.0254	0.0227
	N-t65i11xx	100	0.1384	0.2009	0.1942	**0.1458**	0.1509		tai35b	35	0.1190	0.2379	0.1441	0.0881	**0.0649**
	N-t65l11xx	100	0.1284	0.1791	**0.1075**	0.1327	0.1350		tai40a	40	0.0342	0.0340	**0.0121**	0.0341	0.0313
	N-t65n11xx	100	**0.1438**	0.2125	0.1696	0.1567	0.1597		tai40b	40	0.2517	0.3213	**0.1922**	0.2558	0.2072
	N-t65w11xx	110	0.1823	0.2214	**0.1668**	0.1672	0.1688		tai50a	50	0.0362	0.0371	**0.0127**	0.0376	0.0363
	N-t69r11xx	110	**0.1558**	0.2143	0.2009	0.1607	0.1621		tai50b	50	0.2009	0.2897	**0.0705**	0.1201	0.2739
	N-t70b11xx	110	0.1768	0.2198	0.1724	0.1734	**0.1662**		tai60a	60	**0.0059**	0.0076	0.0066	0.0063	0.0062
	N-t70d11xx	110	0.1785	0.2252	**0.1426**	0.1749	0.1684		tai60b	60	0.1773	0.2295	**0.0438**	0.0997	0.1440
	N-t70d11xxb	110	0.1799	0.2277	**0.1442**	0.1766	0.1822		tai64c	64	0.0378	0.0361	**0.0019**	0.0372	0.0064
	N-t70f11xx	120	0.1837	0.2132	**0.1106**	0.1829	0.1704		tai80a	80	**0.0041**	0.0052	0.0044	0.0051	0.0044
	N-t70i11xx	120	0.1777	0.2236	0.1955	**0.1716**	0.1757		tai80b	80	0.0639	0.0673	**0.0123**	0.0620	0.0529
	N-t70k11xx	120	**0.1843**	0.2246	0.1914	0.1857	0.1845		tai100a	100	0.0301	0.0296	**0.0034**	0.0300	0.0301
	N-t70l11xx	120	0.1589	0.2091	0.2286	0.1650	**0.1544**		tai100b	100	0.1417	0.1504	0.1011	0.1340	**0.0423**
	N-t70n11xx	120	0.1717	0.2278	**0.1183**	0.1818	0.1776		tai150b	150	0.0106	0.0115	-	0.0105	**0.0070**
PFSP	tai50_5_0	50	0.0509	0.1426	0.0393	0.0384	**0.0100**	TSP	burma14	14	0.0765	0.0293	**0.0035**	0.0491	0.0109
	tai50_5_1	50	0.0598	0.1435	0.0472	0.0511	**0.0178**		ulysses16	16	0.0440	0.0145	0.0417	0.0341	**0.0119**
	tai50_5_2	50	0.0469	0.1289	**0.0164**	0.0303	0.0472		gr17	17	0.0766	0.0314	**0.0161**	0.0769	0.0176
	tai50_5_3	50	0.0672	0.1287	**0.0214**	0.0480	0.0628		ulysses22	22	0.1791	0.3612	0.0406	0.0950	**0.0271**
	tai50_5_4	50	0.0450	0.1183	**0.0115**	0.0397	0.0421		gr24	24	0.3268	0.6673	0.1748	0.2424	**0.1385**
	tai50_10_0	50	0.0835	0.1163	**0.0137**	0.0730	0.0399		fri26	26	0.3606	0.7356	**0.0833**	0.2182	0.0887
	tai50_10_1	50	0.0914	0.1296	0.0361	0.0659	**0.0320**		bays29	29	0.5205	0.8591	**0.0993**	0.4162	0.2772
	tai50_10_2	50	0.0813	0.1481	**0.0294**	0.0616	0.0517		dantzig42	42	1.4397	1.5783	**0.0794**	1.4117	1.4195
	tai50_10_3	50	0.0758	0.1269	**0.0092**	0.0732	0.0332		swiss42	42	1.4632	1.4596	**0.0866**	1.4377	1.3774
	tai50_10_4	50	0.0764	0.1210	0.0199	0.0649	**0.0117**		gr48	48	1.5664	1.5779	**0.0807**	1.5546	1.5371
	tai50_20_0	50	0.0728	0.1024	**0.0235**	0.0640	0.0331		hk48	48	1.5937	1.6374	**0.3115**	1.5237	1.4873
	tai50_20_1	50	0.0644	0.0996	0.0365	0.0524	**0.0227**		eil51	51	1.7279	1.7347	**0.0972**	1.7347	1.7080
	tai50_20_2	50	0.0797	0.1163	**0.0120**	0.0614	0.0468		berlin52	52	1.4705	1.5016	**0.0506**	1.4757	1.2737
	tai50_20_3	50	0.0639	0.0939	0.0213	0.0592	**0.0204**		st70	70	1.8370	1.8110	**1.2924**	1.8218	1.7954
	tai50_20_4	50	0.0738	0.1103	**0.0297**	0.0570	0.0417		eil76	76	0.2221	0.2223	**0.0616**	0.2151	0.1953
	tai100_5_0	100	0.0958	0.1192	0.1223	0.0854	**0.0725**		pr76	76	0.2136	0.2129	**0.0429**	0.2126	0.2015
	tai100_5_1	100	0.0922	0.1188	0.1204	0.0916	**0.0773**		gr96	96	3.1000	3.0816	**1.8725**	3.0837	3.0407
	tai100_5_2	100	0.1047	0.1306	0.1022	0.0946	**0.0862**		rat99	99	3.1828	3.1798	**1.9184**	3.1738	3.1281
	tai100_5_3	100	0.0870	0.1359	0.1534	0.0889	**0.0810**		kroA100	100	3.6654	3.6295	**1.8218**	3.6661	3.6289
	tai100_5_4	100	0.0974	0.1376	0.1003	0.0958	**0.0926**		kroC100	100	3.6350	3.5640	**3.3738**	3.6081	3.4630
	tai100_10_0	100	0.1125	0.1320	**0.0908**	0.1135	0.0926		eil101	101	2.5547	2.5353	**1.2587**	2.5464	2.5224
	tai100_10_1	100	0.1273	0.1544	**0.0478**	0.1207	0.1064		pr107	107	4.8875	4.9187	**4.1552**	4.8558	4.8935
	tai100_10_2	100	0.1185	0.1385	0.1010	0.1087	**0.0990**		pr124	124	5.5217	5.5566	**3.7906**	5.4682	5.4870
	tai100_10_3	100	0.0997	0.1217	**0.0498**	0.0988	0.0811		ch130	130	**0.0298**	0.0361	-	0.0304	0.0360
	tai100_10_4	100	0.1233	0.1511	**0.0866**	0.1242	0.0999		pr136	136	0.0241	0.0246	-	**0.0227**	0.0254
	tai100_20_0	100	0.1022	0.1157	**0.0659**	0.0978	0.0807		gr137	137	0.0297	0.0337	-	0.0270	**0.0148**
	tai100_20_1	100	0.1057	0.1117	**0.0674**	0.1017	0.0802		pr144	144	0.0386	0.0379	-	0.0365	**0.0272**
	tai100_20_2	100	0.0946	0.1142	0.0888	0.0985	**0.0786**		kroA150	150	0.0254	0.0356	-	0.0356	**0.0243**
	tai100_20_3	100	0.0997	0.1059	**0.0718**	0.0974	0.0777		ch150	150	0.0188	0.0210	-	0.0198	**0.0213**
	tai100_20_4	100	0.1065	0.1180	0.0909	0.1005	**0.0826**		pr152	152	0.0287	0.0315	-	0.0282	**0.0250**

C. Statistical Testing

In order to state whether there exist statistical differences among the algorithms, we applied the non-parametric Friedman's test to the average ARPD results obtained by M_k, M_c, M_u, GM_k and GM_c for each problem. A level $\alpha = 0.05$ of significance was set. The statistical test reported significant differences among the algorithms in all the problems. Therefore, a post-hoc method was used to carry out all the pairwise comparisons and determine which algorithms are the best performing ones. In particular, Shaffer's static procedure is used, as suggested for such cases in [43]. Again, the significance level was fixed to $\alpha = 0.05$. In order to avoid noise in the statistical test, instances with missing values of M_u were not considered. Results of the statistical test are summarised as critical difference diagrams in Fig. 1.

The statistical analysis reveals that, except for the TSP, there is not just one algorithm that performs the best in the problems. However, critical difference diagrams show that M_u is the most stable algorithm, being always ranked first or second. Alternatively, M_k and M_c are the algorithms

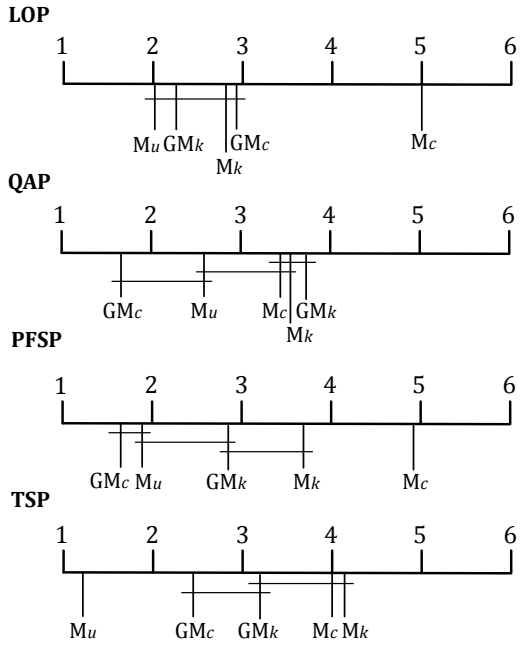

Fig. 1: Critical difference ranking diagrams of the results.

that behave the worst, especially M_c. In addition, the test shows that in the particular case of the PFSP the EDAs have very different behaviours, finding 7 pairwise comparisons statistically different out of 10. Inversely, in the LOP the algorithms performed similarly, with only four comparisons being statistically significant. As a final remark, it is worth mentioning that GMM outperforms MM, when using the same metric, almost systematically.

VI. Conclusions & Future Work

In this paper we extend the use of Mallows and Generalized Mallows distance-based ranking models, in estimation of distribution algorithms. Beyond the commonly used Kendall's-τ distance, two new distance metrics, Cayley and Ulam, have been introduced. In order to analyse their performance when solving permutation-based combinatorial optimisation problems, a benchmark of 120 instances of four well known problems was proposed.

The conducted experiments demonstrated that there is not just one EDA that always performs the best. However, the statistical analysis revealed that M_u is the most stable EDA among the compared approaches. Alternatively, the results confirmed that Generalized Mallows EDAs are preferred to the Mallows EDAs under the same distance, which is quite obvious taking into account that GMM uses n parameters to calculate the probability distribution, and the MM only 2.

As future work, there are many trends that deserve further study. On the one hand, the experimental study showed that the newly introduced Cayley and Ulam distances are able to outperform the Kendall's-τ-based EDAs. Particularly, M_u is the most competitive proposal for the LOP and the TSP, while GM_c is preferred for the QAP and the PFSP. We think that the outstanding performance of these two EDAs could be motivated by the number of permutations that Cayley and Ulam consider at a given distance, being significantly larger than for Kendall's-τ. This aspect could influence the exploration/exploration abilities of the EDA.

On the other hand, as investigated in [44], it could be interesting to analyze the relation between Mallows and Generalized Mallows EDAs, and the neighborhood system induced by the distance metrics studied in this paper.

Finally, taking into account the large performance variations observed for the studied algorithms, new EDA solutions that combine different distance metrics during the search should be investigated.

References

[1] S. Tsutsui and G. Wilson, "Solving Capacitated Vehicle Routing Problems Using Edge Histogram Based Sampling Algorithms," in *Proceedings of the IEEE Conference on Evolutionary Computation, Portland, Oregon (USA)*, 2004, pp. 1150–1157.

[2] S. R. Hejazi and S. Saghafianz, "Flowshop-scheduling problems with makespan criterion: a review," 2004.

[3] R. E. Burkard, E. Çela, P. M. Pardalos, and L. S. Pitsoulis, "The quadratic assignment problem," 1998.

[4] C. Wang, C. Chu, and J.-M. Proth, "A branch-and-bound algorithm for n-job two machine flow shop scheduling problems," in *Emerging Technologies and Factory Automation, 1995. ETFA '95, Proceedings., 1995 INRIA/IEEE Symposium on*, vol. 2, oct 1995, pp. 375 –383 vol.2.

[5] F. Glover, G. Gutin, A. Yeo, and A. Zverovich, "Construction heuristics and domination analysis for the asymmetric tsp," *European Journal of Operational Research*, vol. 129, pp. 85–94, 2001.

[6] T. Stützle, "Iterated local search for the quadratic assignment problem," *European Journal of Operational Research*, vol. 174, no. 3, pp. 1519–1539, 2006.

[7] P. Larrañaga, C. M. H. Kuijpers, R. Murga, I. Inza, and S. Dizdarevic, "Genetic algorithms for the travelling salesman problem: A review of representations and operators," *Artificial Intelligence Review*, vol. 13, pp. 129–170, 1999.

[8] C. Chira, C. M. Pintea, G. C. Crisan, and D. Dumitrescu, "Solving the linear ordering problem using ant models," in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, ser. GECCO '09. New York, NY, USA: ACM, 2009, pp. 1803–1804.

[9] M. F. Tasgetiren, Y.-C. Liang, M. Sevkli, and G. Gencyilmaz, "A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem," *European Journal of Operational Research*, vol. 177, no. 3, pp. 1930 – 1947, 2007.

[10] J. Ceberio, E. Irurozki, A. Mendiburu, and J. A. Lozano, "A review on Estimation of Distribution Algorithms in Permutation-based Combinatorial Optimization Problems," *Progress in Artificial Intelligence*, vol. 1, no. 1, pp. 103–117, January 2012.

[11] J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms (Studies in Fuzziness and Soft Computing)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[12] H. Mühlenbein and G. Paaß, "From Recombination of Genes to the Estimation of Distributions I. Binary Parameters," in *Lecture Notes in Computer Science 1411: Parallel Problem Solving from Nature - PPSN IV*, 1996, pp. 178–187.

[13] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2002.

[14] M. Pelikan, K. Sastry, and E. Cantú-Paz, *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications (Studies in Computational Intelligence)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[15] R. Santana, P. Larrañaga, and J. A. Lozano, "Protein folding in simplified models with Estimation of Distribution Algorithms," *IEEE Transactions On Evolutionary Computation*, vol. 12, no. 4, pp. 418–438, 2008.

[16] Q. Zhang, J. Sun, and E. Tsang, "An Evolutionary Algorithm With Guided Mutation for the Maximum Clique Problem," *IEEE Transactions on Evolutionary Computation*, 2005.

[17] J. Ceberio, L. Hernando, A. Mendiburu, and J. A. Lozano, "Understanding instance complexity in the linear ordering problem," in *The International Conference on Intelligent Data Engineering and Automated Learning (IDEAL-2013)*, vol. LNCS 8206. Springer-Verlag, October 2013, pp. 479–486.

[18] R. Santana, P. Larrañaga, and J. A. Lozano, "Combining variable neighborhood search and estimation of distribution algorithms in the protein side chain placement problem," *Journal of Heuristics*, vol. 14, no. 5, pp. 519–547, 2008.

[19] J. Huang, C. Guestrin, and L. Guibas, "Fourier theoretic probabilistic inference over permutations," *Journal of Machine Learning Research (JMLR)*, vol. 10, pp. 997–1070, May 2009.

[20] J. Ceberio, A. Mendiburu, and J. A. Lozano, "Introducing the Mallows Model on Estimation of Distribution Algorithms," in *In Proceedings of International Conference on Neural Information Processing (ICONIP)*, 2011.

[21] 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 Transactions on Evolutionary Computation*, 2013.

[22] J. Ceberio, A. Mendiburu, and J. A. Lozano, "The Plackett-Luce Ranking Model on Permutation-based Optimization Problems," in *IEEE Congress on Evolutionary Computation*, June 2013, pp. 494 – 501.

[23] J. A. Aledo, J. A. Gámez, and D. Molina, "Tackling the rank aggregation problem with evolutionary algorithms," *Applied Mathematics and Computation*, vol. 222, pp. 632–644, 2013.

[24] C. L. Mallows, "Non-null ranking models." *Biometrika*, vol. 44, no. 1-2, pp. 114–130, 1957.

[25] P. Diaconis, *Group representations in probability and statistics*, ser. Institute of Mathematical Statistics Lecture Notes—Monograph Series, 11. Hayward, CA: Institute of Mathematical Statistics, 1988.

[26] M. A. Fligner and J. S. Verducci, "Distance based ranking Models," *Journal of the Royal Statistical Society*, vol. 48, no. 3, pp. 359–369, 1986.

[27] D. E. Critchlow, M. A. Fligner, and J. S. Verducci, "Probability models on rankings," *Journal of Mathematical Psychology*, vol. 35, no. 3, pp. 294 – 318, 1991.

[28] M. Fligner and J. Verducci, "Multistage ranking models," *Journal of the American Statistical Association*, vol. 83, no. 403, pp. 892–901, 1988.

[29] E. Irurozki, B. Calvo, and J. A. Lozano, "Sampling and learning mallows and generalized mallows models under the cayley distance," University of the Basque Country UPV/EHU, Tech. Rep., 2014.

[30] ——, "Sampling and learning the mallows model under the ulam distance," University of the Basque Country UPV/EHU, Tech. Rep., 2014.

[31] R. Martí and G. Reinelt, *The linear ordering problem: exact and heuristic methods in combinatorial optimization*. Springer, 2011, vol. 175.

[32] J. Ceberio, A. Mendiburu, and J. A. Lozano, "The Linear Ordering Problem Revisited," University of the Basque Country UPV/EHU, Tech. Rep., 2014.

[33] K. Baker, *Introduction to sequencing and scheduling*. Wiley, 1974.

[34] D. E. Goldberg and R. L. Jr., "Alleles Loci and the Traveling Salesman Problem," in *ICGA*, 1985, pp. 154–159.

[35] T. C. Koopmans and M. J. Beckmann, "Assignment Problems and the Location of Economic Activities," Cowles Foundation for Research in Economics, Yale University, Cowles Foundation Discussion Papers 4, 1955.

[36] A. Ali and M. Meila, "Experiments with Kemeny ranking: What works when?" *Mathematical Social Sciences*, 2011.

[37] E. Irurozki, B. Calvo, and J. A. Lozano, "An r package for permutations, mallows and generalized mallows models," University of the Basque Country UPV/EHU, Tech. Rep., 2014.

[38] B. Mandhani and M. Meila, "Tractable Search for Learning Exponential Models of Rankings," *Journal of Machine Learning Research (JMLR)*, vol. 5, pp. 392–399, 2009.

[39] W. Cheng and E. Hullermeier, "A Simple Instance-Based Approach to Multilabel Classification Using the Mallows Model," in *Workshop Proceedings of Learning from Multi-Label Data*, Bled, Slovenia, 2009, pp. 28–38.

[40] I. Caragiannis, A. D. Procaccia, and N. Shah, "When Do Noisy Votes Reveal the Truth?" in *Proceedings of the Fourteenth ACM Conference on Electronic Commerce*, ser. EC '13. New York, NY, USA: ACM, 2013, pp. 143–160.

[41] G. Reinelt, "TSPLIB - A t.s.p. library," Universität Augsburg, Institut für Mathematik, Augsburg, Tech. Rep. 250, 1990.

[42] [Online]. Available: http://mistic.heig-vd.ch/taillard/problemes.dir/problemes.html

[43] S. Garcia, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization," *Journal of Heuristics*, vol. 15, no. 6, pp. 617–644, 2009.

[44] C. Echegoyen, A. Mendiburu, R. Santana, and J. A. Lozano, "On the taxonomy of optimization problems under estimation of distribution algorithms," *Evol. Comput.*, vol. 21, no. 3, pp. 471–495, Sep. 2013.