

Balancing the Diversification-Intensification Trade-off Using Mixtures of Probability Models

Joan Alza*, Josu Ceberio* and Borja Calvo*

*Faculty of Computer Science, 20018 Donostia, Spain

University of the Basque Country UPV/EHU

email: jalza002@ikasle.ehu.eus, josu.ceberio@ehu.eus, borja.calvo@ehu.eus

Abstract—The trade-off between diversification and intensification has been investigated recurrently in the field of evolutionary computation. Proof of this is the numerous approaches that have been devoted to finding a balance in the diversification-intensification behavior of algorithms. Despite the large amount of work on this topic, dynamically adjusting such behavior is still difficult and depends on the algorithm at hand. In this paper, we focus on estimation of distribution algorithms (EDAs).

Usually, research on EDAs mainly focuses on the design of probability models that either represent as best as possible the characteristics of the problem, or accurately fit the domain of the solutions. In this work, we propose implementing mixtures of probability models that permit the dynamic adjustment of the scope of the EDA. Particularly, we design a mixture model that combines two unimodal Thurstone family probability models: the Plackett-Luce model and the Bradley-Terry model. The first model tends to concentrate the probability around the mode, while the second spreads the probability more. Using a homogeneity measure on the population of solutions, we dynamically decide the ratio of solutions to sample from each model. Performed experiments on the linear ordering problem demonstrate that this research line is definitively promising.

I. INTRODUCTION

Since the work of Muhlenbein et al. [1], a large amount of research works, both methodological and applied, has been done as regards estimation of distribution algorithms (EDAs) [2], [3], [4], [5]. These algorithms are a type of population-based evolutionary algorithm designed for solving either combinatorial or continuous domain optimization problems. Based on machine learning techniques, at each iteration, EDAs learn a probabilistic model from a set of good quality solutions (trying to explicitly express the most promising features that make solutions good). Subsequently, by sampling the model estimated in the previous step, a new set of solutions is created. The algorithm iterates until a certain stopping criterion is met.

The majority of works on EDAs have followed two research paths: (1) applying existing canonical designs to real-world problems (hybridizing if necessary) or (2) designing and using new probability models that accurately model the domain of solutions. An illustrative example of the second path is the recent research carried out on *permutation-based problems* [6], [7]. The authors proposed introducing probability models that explicitly define a probability distribution on the whole set of permutations that can be efficiently learned and sampled. In this sense, Mallows [8] and Generalized Mallows models [9] showed interesting performance by using a variety of

distances-metrics on permutations that adjust the model to the different problems [10]. Similarly, the Plackett-Luce model was also investigated in the framework of EDAs [11].

As in many evolutionary algorithms, a successful application of EDAs also depends on efficiently controlling the diversification-intensification trade-off. In previous works, this point did not receive significant attention, and the balance in the trade-off was solved by introducing additional procedures: local search, restart mechanisms, constructive algorithms, etc.

In this work, we approach the problem of finding a proper trade-off of the diversification-intensification behavior of the algorithm from its design. Specifically, we propose using a mixture of two different probability models for the domain of permutations: the Plackett-Luce [12], [13], [14] and Bradley-Terry models [15]. The Plackett-Luce model is a parameterized distribution over permutations that, based on a vector of weights, models the preference of elements to appear in the top ranks (initial positions). Similarly, the Bradley-Terry model is also parameterized with a vector of weights, however, it models the pairwise precedence of the elements in the permutation. Despite the slight difference, Plackett-Luce tends to concentrate the probability around the mode with more emphasis than Bradley-Terry, which is more sparse.

The idea is to, based on the homogeneity of the population, dynamically adapt the weight of each of the models in the mixture. When the population is heterogeneous the aim is to intensify, and therefore we give more relevance to the Plackett-Luce model. Conversely, when the population is homogeneous, the goal is to diversify, and therefore Bradley-Terry is more suitable. To that end, we used the Nogueira's index [16] as the weight of the models in the mixture.

It is worth noting that, to our knowledge, there is no efficient sampling method for the Bradley-Terry model. Consequently, in this work we present two novel algorithms for sampling the Bradley-Terry model: an insertion sort algorithm (Heuristic algorithm) and a sorting algorithm comparing two random elements (Metropolis-Hastings algorithm) [17].

With the aim of evaluating the performance of the mixture model, two sets of experiments have been carried out. On the one hand, we show that the Bradley-Terry model sampling techniques are correctly developed. On the other hand, we run the algorithm on a set of instances of the *Linear Ordering Problem* (LOP) [18], and compare the results obtained by the Plackett-Luce EDA and the mixture EDA. The experiments

performed on the LOP clearly show the benefits of using the proposed mixture.

The remainder of the paper is organized as follows: in the next section a background on probability models for permutation spaces based on order statistics is introduced. Next, in Section III, the first contribution is introduced, efficient sampling methods for the Bradley-Terry models. Afterwards, in Section IV, a diversification technique based on a mixture of Plackett-Luce and Bradley-Terry models is proposed. Correctness experiments, and performance analysis experiments are carried out in Section V. Section VI introduces a discussion on the use of mixture models to balance the trade-off. Finally, Section VII concludes the paper.

II. PROBABILITY MODELS FOR PERMUTATION SPACES

A permutation is a bijection function σ of the set $\{1, \dots, n\}$ onto itself. $\sigma(i)$ (also denoted as σ_i) stands for the element at position i , and $\sigma^{-1}(i)$ represents the position of element i . The group of all permutations of size n , is denoted as \mathbb{S}_n , and is also known as the *symmetric group*.

The Bradley-Terry and Plackett-Luce models are probability models defined on \mathbb{S}_n . In the following paragraphs, we describe their characteristics, which are relevant for this work.

The Bradley-Terry model induces a probability model based on the paired comparison between all the elements in the permutation. Specifically, it models the probability of i beating j ($i \prec j$), where i and j are the elements of the permutation. In our case, we say that i beats j when the position of i is smaller than the position of j , i.e., when i appears before j .

Mathematically, given a vector of $\mathbf{w} = \{w_1, \dots, w_n\}$, the probability of pairwise comparison is

$$P(i \prec j) = \frac{w_i}{w_i + w_j} \quad (1)$$

where w_i and w_j denote the weights of elements i and j .

Therefore, the probability of any permutation $\sigma \in \mathbb{S}_n$ (up to a normalization constant) is obtained by extending the previous expression to all pairwise comparisons as

$$P_{BT}(\sigma|\mathbf{w}) \propto \prod_{i=1}^{n-1} \prod_{j=i+1}^n \frac{w_{\sigma(i)}}{w_{\sigma(i)} + w_{\sigma(j)}}. \quad (2)$$

The Plackett-Luce model can be seen as an extension of the Bradley-Terry model, as it extends the comparison that makes the Bradley-Terry model to n elements. Mathematically, given a parameter vector \mathbf{w} , the probability of any $\sigma \in \mathbb{S}_n$ under the Plackett-Luce model is given by

$$P_{PL}(\sigma|\mathbf{w}) = \prod_{i=1}^{n-1} \frac{w_{\sigma(i)}}{\sum_{j=i}^n w_{\sigma(j)}}. \quad (3)$$

Both models are similar, for both, the larger the parameter w_i , the greater the probability that the element i appears in an initial position of the permutation. Indeed, for both models the mode is the same: the permutation that orders the element decreasingly according to the weight vector. However, in the case of Bradley-Terry the weights are used independently for

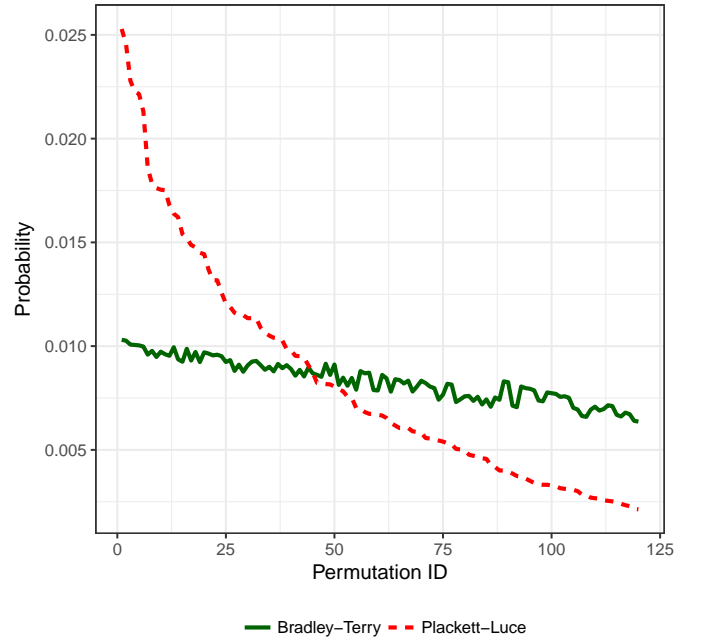


Fig. 1. Comparison of the Bradley-Terry and Plackett-Luce distributions. The x -axis represents the index of each permutation of size 5 and the y -axis the probability under a Bradley-Terry and a Plackett-Luce model, both with the same vector of weights. The permutations are ordered according to the probability under the PL model.

all the pairs of positions while in the case of PL at each position the weight of all the remaining elements is considered. Due to this different use of the weights, the probability distributions represented by both models are different (see Fig. 1). In particular, as can be seen in the figure, Bradley-Terry is more sparse than Plackett-Luce, i.e., the mass of probability is more evenly distributed among the space of permutation. In this work we will exploit this difference in order to balance the diversification and intensification of the search.

As these models are going to be used in the framework of EDAs, it is necessary to provide efficient methods to, given a set of permutations, learn the vector of weights, and to sample new solutions that follow the probability distribution defined by the model. In the case of Plackett-Luce, we consider the learning and sampling methods proposed in [11]. As regards the Bradley-Terry model, learning the maximum likelihood estimators of the weights vector has been reported in [19], and is the method followed in this work. Nonetheless, to our knowledge, there is no method in the literature to sample the Bradley-Terry model. Therefore, in the next section, we propose two methods that, as proved later, allow us to sample the probability distribution efficiently.

III. SAMPLING THE BRADLEY-TERRY MODEL

In this section, we propose two methods to sample the Bradley-Terry model: a Metropolis-Hastings algorithm and a

heuristic algorithm.

A. Metropolis-Hastings algorithm

As a first approach we have implemented a Metropolis-Hastings algorithm [20], which is a generic method to sample probability distributions. It is based on a Markovian random walk, where we move from one element (permutation in our context) to another according to the ratio of probabilities the distribution being sampled (P) assigns to the old and the new elements (see Alg. 1 for more details). This process represents a Markov chain model whose stationary distribution is P and, thus, after a burn-in period of convergence, the sequence produced by the method approximates a sample of P .

Algorithm 1 Metropolis-Hastings sampling

```

 $\sigma^1 \leftarrow$  Random permutation of size  $n$ 
for  $t = 1, 2, \dots, \text{burn-in} + K$  do
   $\sigma' \leftarrow$  Swap two random positions in  $\sigma^t$ 
  With probability  $\min\{\frac{P_{\sigma'}}{P_{\sigma^t}}, 1\}$ 
     $\sigma^{t+1} \leftarrow \sigma'$ 
  otherwise
     $\sigma^{t+1} \leftarrow \sigma^t$ 
end for
return  $\sigma^{\text{burn-in}+1}, \dots, \sigma^{\text{burn-in}+K+1}$ 

```

The problem with this method is that, as it is a random walk, contiguous elements in the sequence of permutations produced are, by definition, very similar. If the sample size is small, in the context of EDAs the reduced variability may lead to the premature convergence of the algorithm. In order to avoid this problem, we will use the algorithm in a different way. Instead of producing one single sample of size K , we will run the algorithm K times, producing a single sample each time. Although it is computationally more expensive, this way it is possible to keep the diversity of the new population.

B. Heuristic algorithm

The Metropolis-Hastings algorithm may be a good approach for small problems, but for increasing sizes it will eventually be prohibitively expensive. For that reason, we have proposed a heuristic algorithm to sample the Bradley-Terry model.

The logic behind the algorithm is to follow the process represented by the Bradley-Terry model, i.e., taking the decisions based on pairwise comparisons. Through a constructive process, we randomly select elements to be introduced in the solution and, based on the probabilities derived from Eq. 1, we select the position where the new element will be inserted. The pseudo code of the heuristic algorithm is presented in Alg. 2.

The stopping criterion shown in the algorithm is set to avoid an infinite loop. It means that the process will stop when we compare the element with an element that it has just been compared with. The following examples clarifies the process:

- 1) Take a random element from $R = \{1, 2, 3\}$, e.g., 3, and insert it into S . Now $S = (3)$

Algorithm 2 Heuristic sampling

```

 $S \leftarrow$  Empty vector
 $S \leftarrow \{1, \dots, n\}$ 
 $k \leftarrow$  Randomly choose an element from  $R$ 
Insert  $k$  into  $S$  and remove from  $R$ 
while  $R \neq \emptyset$  do
   $k \leftarrow$  Choose a random element from  $R$ 
   $z \leftarrow$  Choose a random index of  $S$ 
  while Stopping criterion is not met do
    With probability  $\frac{w_k}{w_k + w_{S(z)}}$ 
       $z \leftarrow z - 1$ 
    Otherwise
       $z \leftarrow z + 1$ 
  end while
   $S \leftarrow$  Insert  $k$  in  $S(z)$ 
   $R \leftarrow$  Remove  $k$  from  $R$ 
end while
 $\sigma \leftarrow S$ 
return  $\sigma$ 

```

- 2) Take a random element from the remaining $R = \{1, 2\}$, e.g., 2, and a random position in S (in this case there is only one, $z = 1$).
 - Compare the element to be inserted with the element in the position selected, i.e., elements 2 and 3. With probability $\frac{w_2}{w_2 + w_3}$, place the new element *before* the selected position. Suppose the result is placing it after, so the result would be $S = (3, 2)$.
- 3) Take a random element from the remaining $R = \{1\}$ (1 is the only option), and a random position in S , e.g., $z = 2$.
 - Compare the new element (1) with the element in the selected position (element 2, position 2). With probability $\frac{w_1}{w_1 + w_2}$, decide to place the new element before the selected position. Suppose the decision is to place it before.
 - Compare the new element (1) with the element in the new position (element 3, position 1). With probability $\frac{w_1}{w_1 + w_3}$, decide to place the new element before the selected position. Suppose the decision is to place it after. Insert the new element in the selected position. Now $S = (3, 1, 2)$
- 4) All the elements are in S , thus $\sigma = S$.

IV. DIVERSIFYING THE SEARCH WITH A MIXTURE OF MODELS IN EDAS

As mentioned in the introduction of the manuscript, the good performance of EDAs, as for most of evolutionary algorithms, depends partially on finding a balance between diversifying and intensifying. Taking that into account, we propose a mixture of two models: the Plackett-Luce model and the Bradley-Terry model. Both models belong to the same family and have the same mode. However, the second is more sparse than the first. Therefore, by balancing the weight of

each of the models at each iteration, we can balance the behavior of the algorithm.

Formally, the probability of a solution $\sigma \in \mathbb{S}_n$ is

$$P(\sigma) = \Theta P_{PL}(\sigma) + (1 - \Theta) P_{BT}(\sigma), \quad (4)$$

where Θ and $1 - \Theta$ are the weights of the models and, P_{PL} and P_{BT} are the Plackett-Luce and Bradley-Terry models respectively.

A critical point in balancing the intensification and diversification is the Θ parameter. It should be a value that ranges between 0 and 1 (or, with finite, known bounds at least). Those limits should represent the two extreme situations, $\Theta = 0$ when the population is highly diverse (e.g., complete random permutations) and $\Theta = 1$ when there is no diversity at all (all the permutations are the same).

As a first approach, we have selected Nogueira's index [16], as it perfectly fits the required features. This is an index proposed to evaluate the consistency of rankings, it is bounded at 1 and is corrected by chance, that is, when applied to a uniform sample of rankings, its expected value is 0.

Given a population of solutions (permutations) X , and viewing it as a matrix of numerical entries where a row is a solution for the problem and X_j makes reference to the j^{th} column, Nogueira's index is calculated as:

$$\Theta = 1 - \frac{\frac{1}{n} \sum_{j=1}^n \text{Var}(X_j)}{V_r}, \quad (5)$$

where n is the size of the permutations, and $V_r = \frac{n^2-1}{12}$ is a constant that depends on n .

According to Eq. 5, when the population X is homogeneous, the variance of X_j -s will be 0 and, thus, Θ will equal to 1. In such a situation, the mixture is actually a Bradley-Terry model, and sampling it we intend to increase the diversity of the population. On the contrary, when the variability of the population is high, Θ is expected to decrease, giving more weight to the Plackett-Luce and, therefore, reinforcing the intensification behavior of the algorithm.

Obviously, when compared to other EDAs that use one model, the computational cost in our approach will increase as two models are estimated at each iteration. In Alg. 3, the overall procedure of the mixture EDA is introduced.

Algorithm 3 Pseudo-code of the mixture EDA

```

 $D_0 \leftarrow$  Generate  $M$  random individuals.
 $t \leftarrow 1$ 
repeat
   $D_{t-1} \leftarrow$  Evaluate individuals.
   $D_{t-1}^{Se} \leftarrow$  Choose  $N \leq M$  individuals from  $D_{t-1}$ .
   $w_{PL} \leftarrow$  Estimate the vector of weights of Plackett-Luce from  $D_{t-1}^{Se}$ .
   $w_{BT} \leftarrow$  Estimate the vector of weights of Bradley-Terry from  $D_{t-1}^{Se}$ .
   $\Theta \leftarrow$  Calculate Nogueira's index.
   $X^t \leftarrow$  Sample  $M - 1$  individuals from Eq. 4.
until Stopping criterion is met

```

The stopping criterion is set to a maximum number of iterations.

V. EXPERIMENTAL STUDY

In this section, we carry out two different types of experiments. On the one hand, in order to check the correctness of the sampling methods developed for the Bradley-Terry model, we perform some recovery tests. On the other hand, in order to evaluate the performance of the proposed algorithm, we run a set of benchmark experiments with the mixture EDA and the Plackett-Luce EDA proposed in [11].

A. Recovery tests

This experiment consists of the following steps. Firstly, we artificially create a Bradley-Terry model by sampling the vector of weights from a uniform Dirichlet distribution. Secondly, the model is sampled in order to obtain a set of solutions. Finally, based on that set of solutions, the vector of weights of the model is estimated. The idea is that, if the learning and sampling methods are correctly designed, the initial vector of parameters should be recovered with the sampled set of solutions.

For the sake of measuring the similarity between the initial and the obtained model, we calculate the Kullback-Leibler divergence between the two distributions.

We consider solutions of size $n = 5$, both sampling methods (heuristic and Metropolis-Hastings), and the experiment was performed for six different samples of solutions: 10, 50, 100, 500, 1000 and 10000. As the sample size increases the divergence should decrease. In the case of Metropolis-Hastings, the *burn-in* period was set to 1000. With the aim of analyzing the variance in the divergence, the experiment was repeated 10 times, and results are presented in Fig. 2 as box-plots.

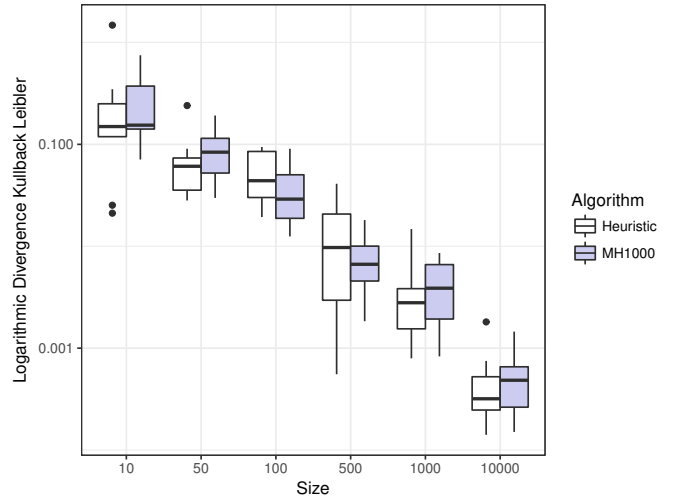


Fig. 2. Kullback-Leibler divergences between the artificially generated distribution and the recovered distribution for the two sampling methods.

As can be observed in Fig. 2, the increment of the sample size produces the reduction of the divergence, being close to

zero for 10000 samples. This last result shows that the model is recovered successfully, and therefore the sampling methods create solutions that follow the true probability distribution. Moreover, for the largest sample size, we can see that the divergence when using the heuristic algorithm is slightly smaller. Taking this into account, and also that the complexity of the heuristic is much lower, for the performance evaluation of the EDA, this method is used to sample the model.

B. Performance evaluation

For the sake of analyzing the performance of the mixture EDA (PL+BT), we compared it with that of the Plackett-Luce EDA (PL). For illustrative purposes, we applied both approaches over the linear ordering problem (LOP) [18], [21]. We chose 50 instances from LOLIB benchmark with sizes that range between $n = 44$ and $n = 79$ ¹.

The parameter setting employed in both EDAs was the following:

- Population size is set to $10n$.
- Truncation selection is applied by choosing the best 10% of individuals.
- Elitism criteria is used.
- $10n$ individuals are sampled at each generation.
- The stopping criteria is set to perform a maximum number of $10000n$ evaluations.

As regards the Plackett-Luce model, a maximum of 1000 iterations were set for the Minorization-Maximization algorithm used to estimate the MLE parameters.

Each algorithm was executed 10 times on each instance. The results are summarized in Table I as the average relative percentage deviation (ARPD) of the median solution found across the repetitions by algorithms. The ARPD is a performance measure of the algorithm and it is calculated as

$$ARPD = \frac{\text{Best known} - \text{Median solution}}{\text{Best known}}. \quad (6)$$

As can be observed, PL+BT outperforms PL in 34 instances out of 50. In one case, both algorithms obtained an equal result, and in the rest of the instances, 15 out of 50, PL was the preferred option.

In order to statistically assess the results obtained, we have followed the Bayesian approach presented in [22], as it provides a deeper insight of the results than the classical statistical tests. In particular, as we cannot assume that the experimental results come from a Gaussian distribution, we have used the Bayesian equivalent of the Wilcoxon's test².

The Bayesian analysis was conducted on the median ARPD obtained by each algorithm in the 10 repetitions. The procedure used requires the definition of what is understood as 'practical equivalence' or 'rope' [22]. In our case we have considered that both approaches are equivalent when the

¹The instances were downloaded from *Input/Output* matrices from LOLIB web page (<http://www.opticom.es/lolib/>).

²We have used the implementation available in the development version of the **scmamp** R package [23] available at <https://github.com/b0rxa/scmamp>.

TABLE I
RESULTS OF THE MIXTURE EDA AND PLACKETT-LUCE EDA FOR THE LOP INSTANCES. THE MEDIAN RESULTS FOUND ACROSS THE 10 REPETITIONS, AND THE RESPECTIVE ARPD TO THE BEST KNOWN OBJECTIVE VALUES ARE REPORTED. RESULTS IN BOLD HIGHLIGHT THE ALGORITHM THAT OBTAINED THE LOWEST ARPD.

Size	Instance	Best Known	PL+BT		PL	
			ARPD	Fitness	ARPD	Fitness
50	N-be75eec	236464	0.007	234805	0.007	234838
50	N-be75np	716994	0.004	714009	0.003	714780
50	N-be75oi	111171	0.004	110759	0.003	110830
50	N-be75tot	980516	0.003	977617	0.004	976464
60	N-stabu70	362512	0.011	358531	0.011	358630
60	N-stabu74	541393	0.008	537220	0.011	535673
60	N-stabu75	553303	0.009	548417	0.012	546589
44	N-t59b11xx	209320	0.001	209065	0.002	208890
44	N-t59d11xx	147354	0.000	147354	0.000	147295
44	N-t59f11xx	122520	0.001	122376	0.001	122391
44	N-t59i11xx	8261545	0.000	8260443	0.001	8255600
44	N-t59n11xx	20928	0.003	20928	0.002	20881
44	N-t65b11xx	356758	0.002	355954	0.004	355278
44	N-t65d11xx	237739	0.004	236717	0.006	236195
44	N-t65f11xx	217295	0.005	216188	0.006	216019
44	N-t65i11xx	14469163	0.001	14449137	0.002	14435204
44	N-t65l11xx	16719	0.002	16693	0.001	16695
44	N-t65n11xx	32157	0.003	32076	0.003	32057
44	N-t65w11xx	138181029	0.003	137702570	0.005	137446550
44	N-t69r11xx	771149	0.004	768155	0.002	769793
44	N-t70b11xx	528419	0.002	527278	0.002	527109
44	N-t70d11xx	376725	0.002	375891	0.006	374369
44	N-t70d11xxb	366469	0.001	366027	0.003	365377
44	N-t70f11xx	360336	0.003	359265	0.004	359036
44	N-t70i11xx	24785782	0.002	24733979	0.002	24724658
44	N-t70k11xx	60659200	0.000	60629900	0.003	60486900
44	N-t70l11xx	25253	0.000	25241	0.000	25241
44	N-t70n11xx	52704	0.004	52503	0.003	52532
44	N-t70u11xx	21716400	0.001	21693500	0.001	21685400
44	N-t70w11xx	224319954	0.003	223741686	0.002	223854505
44	N-t70x11xx	283808865	0.002	283371015	0.003	282916730
44	N-t74d11xx	566089	0.002	564928	0.002	564957
44	N-t75d11xx	578304	0.004	576086	0.004	576006
44	N-t75e11xx	2739219	0.002	2733914	0.004	2729043
44	N-t75i11xx	63567735	0.001	63487099	0.001	63485561
44	N-t75k11xx	108844	0.002	108677	0.002	108601
44	N-t75n11xx	93777	0.001	93653	0.004	93439
44	N-t75u11xx	52708323	0.001	52646880	0.001	52638985
56	N-tiw56n54	91554	0.005	91097	0.004	91173
56	N-tiw56n58	125224	0.003	124826	0.006	124423
56	N-tiw56n62	176715	0.004	176049	0.005	175749
56	N-tiw56n66	226547	0.009	224423	0.007	225027
56	N-tiw56n67	226033	0.005	224800	0.007	224438
56	N-tiw56n72	365146	0.007	362430	0.009	361856
56	N-tiw56r54	102948	0.004	102574	0.005	102437
56	N-tiw56r58	129568	0.005	128892	0.005	128908
56	N-tiw56r66	209491	0.008	207919	0.008	207727
56	N-tiw56r67	222810	0.005	221667	0.008	221107
56	N-tiw56r72	270663	0.007	268687	0.009	268234
79	N-usa79	1813986	0.018	1782047	0.014	1787757

difference in ARPD is smaller than 5×10^{-43} . Fig. 1 shows a summary of the results in a Simplex plot.

Briefly, the points in the plot represent a sampling of the posterior distribution of the probability of win-lose-tie. In other words, the closer a point is to the PL+BT vertex of the triangle (or, equivalently, to the BT or the Rope vertices), the more probable that it is PL+BT produces better results (or equivalently, BT or both algorithms being equal). Therefore, the three areas delimited by the dashed lines show the dominance regions, i.e., the area where the highest probability corresponds to its vertex. For more details, see [22].

The plot shows that there is some uncertainty about the results (due to the sample size, as in this preliminary work we

³We can set this rope at any reasonable value. In our case, we have fixed its value according to the magnitude of the ARPD produced by both algorithms, and mainly with visualization purposes.

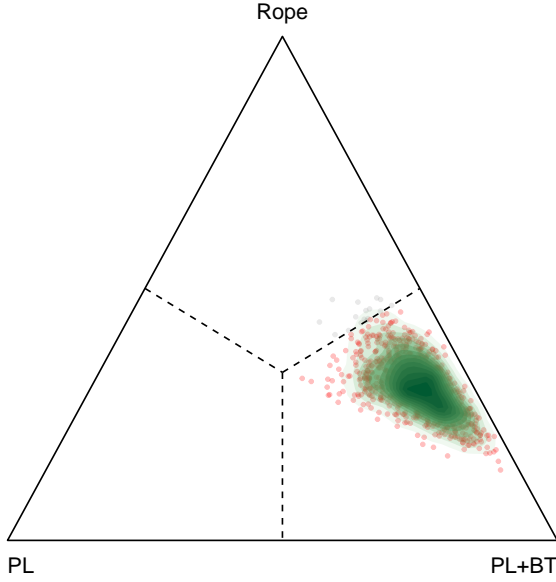


Fig. 3. Simplex plot representing the posterior distribution of win-tie-loose probabilities

have used only 50 instances), but most of the probability mass of the posterior is on the side of the PL+BT, with only some probability for the rope. We can summarize the distribution shown in the plot estimating the expected probability for each situation (PL+BT being better, equal or worse than PL). In that case, the expected probabilities are 0.590 that PL+BT obtains better results, 0.305 that the results of both algorithms are equivalent and 0.104 that PL obtains better results. If we shrink the rope to both algorithms obtaining exactly the same results the expected probabilities are 0.772 for PL+BT, 0.226 for PL and 0.002 for the rope. In other words, it is almost 6 times more probable that PL+BT outperforms PL than the opposite (3.5 times when we only consider the algorithms equal when their median values are exactly equal).

VI. DISCUSSION

From the results, we can conclude that the proposed mixture of models performs better than the Plackett-Luce alone, suggesting that indeed our approach is able to better cope with the diversification-intensification duality of the search. However, the average results presented in the table are not useful to analyze how the mixture model works. In order to see the evolution of the Plackett-Luce model and the mixture, we have plotted the progress of both algorithms together with the value estimated for the Θ at each generation.

Fig. 4 shows an example of such plots. In both plots the x -axis is the same (the number of evaluations). The upper plot shows the fitness values (in logarithmic scale) of the best results obtained by each algorithm while the lower plot shows the value of the Θ . In these plots, we can clearly see

the expected behavior. When the Θ increases, more solutions are sampled from the Bradley-Terry model, this increases the diversity of the population what, in turn it decreases the Θ parameter, and leads to ‘jumps’ in the evolution of the algorithm.

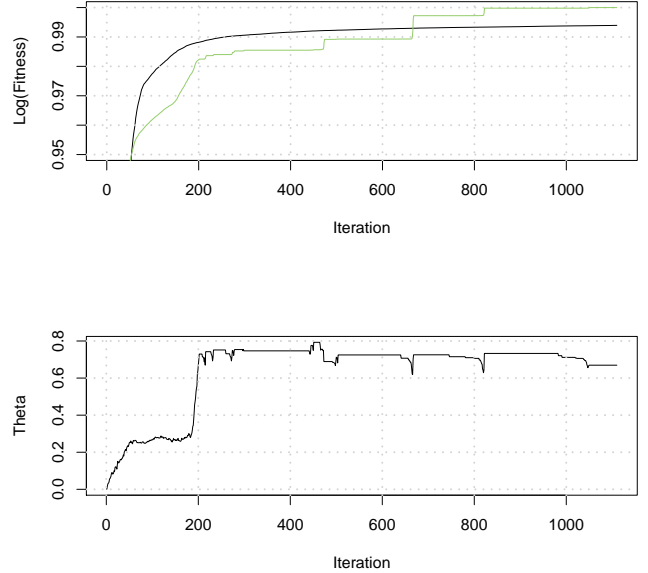


Fig. 4. In the plot at top, the best fitness obtained by PL (in black) and by PL+BT (in green) in one execution on the instance *N-be75oi* is shown. In the plot below, the values obtained for the Nogueira’s index in the same execution are shown.

It is important to point out that the use of the models is probabilistic, i.e., Θ represents the **probability** of sampling from the Plackett-Luce or the Bradley-Terry model. On the other hand, sampling from the Bradley-Terry does not necessarily increase the diversity of the population. For that, we need to sample solutions which are good enough to be included in the new population. That explains the steps observed in the plot. The population gets stuck but, as we are sampling from the Bradley-Terry model, eventually we obtain new solutions which are good enough to be included in the new population.

The result is quite illustrative (similar plots are obtained for other instances). While the Plackett-Luce model converges to a population where improving the solution is hard, the proposed mixture improves the solution quickly at the beginning (where the Plackett-Luce model dominates) but then slows down the convergence as the Bradley-Terry is used more often. These plots show that our approach has the potential to improve the trade-off between diversification and intensification. However, the equilibrium between both is not easy to define. In our proposal we have used Nogueira’s index as the metric to balance both strategies and, although we can clearly see that it slows the convergence of the algorithm, there is still room to test other approaches.

VII. CONCLUSIONS

The trade-off between diversification-intensification has been topical in the field of evolutionary computation, and also in the case of EDAs for which different strategies have been followed in order to find a balance.

In this paper, we propose using a mixture model that facilitates adjusting the diversification-intensification behavior of the algorithm automatically during the optimization processes. In this sense, we divided the work in two: (1) we developed two sampling algorithms for the Bradley-Terry model, and (2), we implemented an EDA that uses a mixture of Plackett-Luce and Bradley-Terry models with the aim of balancing the trade-off between diversification-intensification.

As regards the sampling of the Bradley-Terry model, we have proposed two algorithms (Metropolis-Hastings and Heuristic), and we have compared them in the experimental study. It revealed that both the performance and the computational cost of the heuristic is better than its counterpart.

In order to automatically adjust the behavior of the EDA, we used the Nogueira's index to measure the homogeneity of the population, and decide the ratio of solutions to sample from each model. The conducted experiments pointed out that the mixture EDA obtains better results than the Plackett-Luce EDA. Moreover, the evolution of the search shows that there is a direct relation between the evolution of the Θ parameter and the jumps in the search.

As previously mentioned, this is a preliminary work that shows the potential of using mixtures of models in automatically balancing the intensification and the diversification of the search. However, there are many aspects to be considered in the future. The first and most important is the metric used to define the mixture. In this work we have used Nogueira's index, but other approaches may lead to a better balance. Regarding the distributions, we have focused our attention on the Bradley-Terry/Plackett-Luce pair, as they have the same mode but different variance. However, even if the Bradley-Terry is more sparse than the Plackett-Luce, when learned from an almost uniform population it may be not sparse enough. A possible line for future work is analyzing the use of models with a higher variability. Finally, in this work we have considered only the LOP. The same idea could be extended to other permutation problems and to problems based on other codifications.

ACKNOWLEDGMENT

This work has been partially supported by the Research Groups 2013-2018 (IT-609-13) programs (Basque Government), TIN2016-78365-R (Spanish Ministry of Economy, Industry and Competitiveness).

REFERENCES

- [1] H. Mühlenbein, T. Mahnig, and A. O. Rodriguez, "Schemata, distributions and graphical models in evolutionary optimization," *Journal of Heuristics*, vol. 5, no. 2, pp. 215–247, 1999. [Online]. Available: <http://dx.doi.org/10.1023/A:1009689913453>
- [2] P. Larrañaga and J. A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation*. Springer Science & Business Media, 2001, vol. 2.
- [3] J. A. Lozano, *Towards a new evolutionary computation: advances on estimation of distribution algorithms*. Springer Science & Business Media, 2006, vol. 192.
- [4] M. Pelikan and D. E. Goldberg, "Hierarchical problem solving and the bayesian optimization algorithm," in *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*. Morgan Kaufmann Publishers Inc., 2000, pp. 267–274.
- [5] M. Pelikan, K. Sastry, and E. Cantú-Paz, *Scalable optimization via probabilistic modeling: From algorithms to applications*. Springer, 2007, vol. 33.
- [6] J. Ceberio, *Solving Permutation Problems with Estimation of Distribution Algorithms and Extensions Thereof*. Lambert Academic Publishing, 2015.
- [7] J. Ceberio, E. Irurozki, A. Mendiburu, and J. A. Lozano, "A distance-based ranking model estimation of distribution algorithm for the flow-shop scheduling problem," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 286–300, 2014.
- [8] C. L. Mallows, "Non-null ranking models. i," *Biometrika*, vol. 44, no. 1/2, pp. 114–130, 1957.
- [9] E. Irurozki, "Sampling and learning distance-based probability models for permutation spaces," Ph.D. dissertation, Faculty of Computer Science, University of the Basque Country, November 2014.
- [10] J. Ceberio, "Solving permutation problems with estimation of distribution algorithms and extensions thereof," Ph.D. dissertation, Faculty of Computer Science, University of the Basque Country, December 2014.
- [11] J. Ceberio, A. Mendiburu, and J. A. Lozano, "The plackett-luce ranking model on permutation-based optimization problems," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2013, pp. 494–501.
- [12] J. I. Marden, *Analyzing and Modeling Rank Data*. Chapman & Hall, 1995.
- [13] R. L. Plackett, "The Analysis of Permutations," *Journal of the Royal Statistical Society*, vol. 24, no. 10, pp. 193–202, 1975.
- [14] D. Luce R., *Individual Choice Behavior*. New York: Wiley, 1959.
- [15] P. Bratley, B. L. Fox, and L. E. Schrage, *A Guide to Simulation (2Nd Ed.)*. New York, NY, USA: Springer-Verlag New York, Inc., 1986.
- [16] S. Nogueira, K. Sechidis, and G. Brown, "On the use of spearman's rho to measure the stability of feature rankings," in *Pattern Recognition and Image Analysis*, 2017, pp. 381–391.
- [17] I. Yildirim, "Bayesian inference: Metropolis-hastings sampling," *Dept. of Brain and Cognitive Sciences, Univ. of Rochester, Rochester, NY*, 2012.
- [18] R. Martí and G. Reinelt, *The linear ordering problem: exact and heuristic methods in combinatorial optimization*. Springer Science & Business Media, 2011, vol. 175.
- [19] H. Turner and D. Firth, "Bradley-terry models in R: The BradleyTerry2 package," *Journal of Statistical Software*, vol. 48, no. 9, pp. 1–21, 2012. [Online]. Available: <http://www.jstatsoft.org/v48/i09/>
- [20] W. K. Hastings, *Biometrika*, no. 1, pp. 97–109.
- [21] J. Ceberio, A. Mendiburu, and J. A. Lozano, "The Linear Ordering Problem Revisited," *European Journal of Operational Research*, vol. 241, no. 3, pp. 686–696, 2014.
- [22] A. Benavoli, G. Corani, J. Demšar, and M. Zaffalon, "Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis," *Journal of Machine Learning Research*, vol. 18, no. 77, pp. 1–36, 2017.
- [23] B. Calvo and G. Santafe, "scmamp: Statistical comparison of multiple algorithms in multiple problems," *The R Journal*, vol. 8, no. 1, pp. 248–256, 2015.