Introducing the Mallows Model on Estimation of Distribution Algorithms

Josu Ceberio, Alexander Mendiburu, and Jose A. Lozano

Intelligent Systems Group
Faculty of Computer Science
The University of The Basque Country
Manuel de Lardizabal pasealekua, 1
20018 Donostia - San Sebastian, Spain
jceberio001@ikasle.ehu.es,
{alexander.mendiburu,ja.lozano}@ehu.es
http://www.sc.ehu.es/isg

Abstract. Estimation of Distribution Algorithms are a set of algorithms that belong to the field of Evolutionary Computation. Characterized by the use of probabilistic models to learn the (in)dependencies between the variables of the optimization problem, these algorithms have been applied to a wide set of academic and real-world optimization problems, achieving competitive results in most scenarios. However, they have not been extensively developed for permutation-based problems. In this paper we introduce a new EDA approach specifically designed to deal with permutation-based problems. In this paper, our proposal estimates a probability distribution over permutations by means of a distance-based exponential model called the Mallows model. In order to analyze the performance of the Mallows model in EDAs, we carry out some experiments over the Permutation Flowshop Scheduling Problem (PFSP), and compare the results with those obtained by two state-of-the-art EDAs for permutation-based problems.

Keywords: Estimation of Distribution Algorithms, Probabilistic Models, Mallows Model, Permutations, Flow Shop Scheduling Problem.

1 Introduction

Estimation of Distribution Algorithms (EDAs) [10, 15, 16] are a set of Evolutionary Algorithms (EAs). However, unlike other EAs, at each step of the evolution, EDAs learn a probabilistic model from a population of solutions trying to explicitly express the interrelations between the variables of the problem. The new offspring is then obtained by sampling the probabilistic model. The algorithm stops when a certain criterion is met, such as a maximum number of generations, homogeneous population, or lack of improvement in the last generations.

Many different approaches have been given in the literature to deal with permutation problems by means of EDAs. However, most of these proposals

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 461–470, 2011.

are adaptations of classical EDAs designed to solve discrete or continuous domain problems. Discrete domain EDAs follow the path-representation codification [17] to encode permutation problems. These approaches learn, departing from a dataset of permutations, a probability distribution over a set $\Omega = \{0, \ldots, n-1\}^n$, where $n \in \mathbb{N}$. Therefore, the sampling of these models has to be modified in order to provide permutation individuals. Algorithms such as Univariate Marginal Distribution Algorithm (UMDA), Estimation of Bayesian Networks Algorithm (EBNAs), or Mutual Information Maximization for Input Clustering (MIMIC) have been applied with this encoding to different problems [2, 11, 17].

Adaptations of continuous EDAs [3, 11, 17] use the Random Keys representation [1] to encode a solution with random numbers. These numbers are used as sort keys to obtain the permutation. Thus, to encode a permutation of length n, each index in the permutation is assigned a value (key) from some real domain, which is usually taken to be the interval [0, 1]. Subsequently, the indexes are sorted using the keys to get the permutation. The main advantage of random keys is that they always provide feasible solutions, since each encoding represents a permutation. However, solutions are not processed in the permutation space, but in the largely redundant real-valued space. For example, for length 3 permutation, strings (0.2, 0.1, 0.7) and (0.4, 0.3, 0.5) represent the same permutation (2, 1, 3).

The limitations of these direct approaches, both in the discrete and continuous domains, encouraged the research community of EDAs to implement specific algorithms for solving permutation-based problems. Bosman and Thierens introduced the ICE [3, 4] algorithm to overcome the bad performance of Random Keys in permutation optimization. The ICE replaces the sampling step with a special crossover operator which is guided by the probabilistic model, guaranteeing feasible solutions.

In [18] a new framework for EDAs called Recursive EDAs (REDAs) is introduced. REDAs is an optimization strategy that consists of separately optimizing different subsets of variables of the individual.

Tsutsui et al. [19, 20] propose two new models to deal with permutation problems. The first approach is called Edge Histogram Based Sampling Algorithm (EHBSA). EHBSA builds an Edge Histogram Matrix (EHM), which models the edge distribution of the indexes in the selected individuals. A second approach called Node Histogram Based Sampling Algorithm (NHBSA), introduced later by the authors, models the frequency of the indexes at each absolute position in the selected individuals. Both algorithms simulate new individuals by sampling the marginals matrix. In addition, the authors proposed the use of a template-based method to create new solutions. This method consists of randomly choosing an individual from the previous generation, dividing it into c random segments and sampling the indexes for one of the segments, leaving the remaining indexes unchanged. A generalization of these approaches was given by Ceberio et al. [6], where the proposed algorithm learns k-order marginal models.

As stated in [5], Tsutsui's EHBSA and NHBSA approaches yield the best results for several permutation-based problems, such as Traveling Salesman Problem, Flow Shop Scheduling Problem, Quadratic Assignment Problem or Linear Ordering Problem. However, these approaches are still far from achieving optimal solutions, which means that there is still room for improvement. Note that the introduced approaches do not estimate a probability distribution over the space of permutations that allow us to calculate the probability of a given solution in a closed form. Motivated by this issue and working in that direction, we present a new EDA which models an explicit probability distribution over the permutation space: the Mallows EDA.

The remainder of the paper is as follows: Section 2 introduces the optimization problem we tackle: The Permutation Flow Shop Scheduling Problem. In Section 3 the Mallows model is introduced. In section 4, some preliminary experiments are run to study the behavior of the Mallows EDA. Finally, conclusions are drawn in Section 5.

2 The Permutation Flowshop Scheduling Problem

The Flowshop Scheduling Problem [9] consists of scheduling n jobs $(i = 1, \ldots, n)$ with known processing time on m machines $(j = 1, \ldots, m)$. A job consists of m operations and the j-th operation of each job must be processed on machine j for a specific 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. If the jobs are processed in the same order on different machines, the problem is named as Permutation Flowshop Scheduling Problem (PFSP). The objective of the PFSP is to find a permutation that achieves a specific criterion such as minimizing the total flow time, the makespan, etc. The solutions (permutations) are denoted as $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ where σ_i represents the job to be processed in the i-th position. For instance, in a problem of 4 jobs and 3 machines, the solution $(2, 3, 1, 4)$, indicates that job 2 is processed first, next job 3 and so on.

Let $p_{i,j}$ denote the processing time for job i on machine j, and $c_{i,j}$ denote the completion time of job i on machine j. Then, $c_{\sigma_i,j}$ is the completion time of the job scheduled in the i-th position on machine j. $c_{\sigma_i,j}$ is computed as $c_{\sigma_i,j} = p_{\sigma_i,j} + max\{c_{\sigma_i,j-1}, c_{\sigma_{i-1},j}\}$. As this paper addresses the makespan performance measure, the objective function F is defined as follows:

$$F(\sigma_1, \sigma_2, \ldots, \sigma_n) = c_{\sigma_n,m}$$

As can be seen, the solution of the problem is given by the processing time of the last job σ_n in the permutation, since this is the last job to finish.

3 The Mallows Model

The Mallows model [12] is a distance-based exponential probability model over permutation spaces. Given a distance d over permutations, it can be defined by two parameters: the central permutation σ_0, and the spread parameter θ. (1) shows the explicit form of the probability distribution over the space of permutations:

$$P(\sigma) = \frac{1}{\psi(\theta)} e^{-\theta d(\sigma,\sigma_0)} \tag{1}$$

where $\psi(\theta)$ is a normalization constant. When $\theta > 0$, the central permutation σ_0 is the one with the highest probability value and the probability of the other $n! - 1$ permutations exponentially decreases with the distance to the central permutation (and the spread parameter θ). Because of these properties, the Mallows distribution is considered analogous to the Gaussian distribution on the space of permutations (see Fig. 1). Note that when θ increases, the curve of the probability distribution becomes more peaked at σ_0.

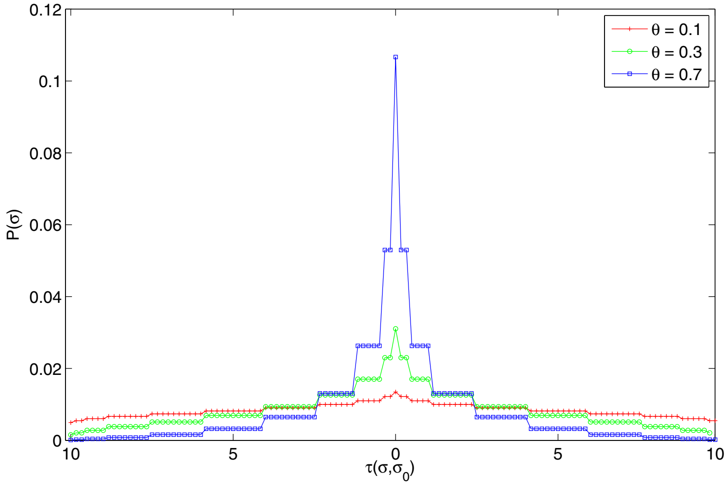

Fig. 1. Mallows probability distribution with the Kendall-τ distance for different spread parameters. In this case, the dimension of the problem is $n = 5$.

3.1 Kendall-τ Distance

The Mallows model is not tied to a specific distance. In fact, it has been used with different distances in the literature such as Kendall, Cayley or Spearman [8]. For the application of the Mallows model in EDAs, we have chosen the Kendall-τ distance. This is the most commonly used distance with the Mallows model, and in addition, its definition resembles the structure of a basic neighborhood system in the space of permutations. Given two permutations σ_1 and σ_2, the Kendall-τ distance counts the total number of pairwise disagreements between both of them i.e., the minimum number of adjacent swaps to convert σ_1 into σ_2. Formally, it can be written as

$$\tau(\sigma_1,\sigma_2) = |\{(i,j) : i < j, (\sigma_1(i) < \sigma_1(j) \wedge \sigma_2(i) > \sigma_2(j)) \\ \vee (\sigma_2(i) < \sigma_2(j) \wedge \sigma_1(i) > \sigma_1(j))\}|.$$

The above metric can be equivalently written as

$$\tau(\sigma_1,\sigma_2) = \sum_{j=1}^{n-1} V_j(\sigma_1,\sigma_2) \tag{2}$$

where $V_j(\sigma_1,\sigma_2)$ is the minimum number of adjacent swaps to set in the j-th position of σ_1, $\sigma_1(j)$, the value $\sigma_2(j)$. This decomposition allows to factorize the distribution as a product of independent univariate exponential models[14], one for each V_j and that (see (3) and (4)).

$$\psi(\theta) = \prod_{j=1}^{n-1} \psi_j(\theta) = \prod_{j=1}^{n-1} \frac{1-e^{-(n-j+1)\theta}}{1-e^{-\theta}} \tag{3}$$

$$P(\sigma) = \frac{e^{-\theta \sum_{j=1}^{n-1} V_j(\sigma,\sigma_0)}}{\prod_{j=1}^{n-1} \psi_j(\theta)} = \prod_{j=1}^{n-1} \frac{e^{-\theta V_j(\sigma,\sigma_0)}}{\psi_j(\theta)} \tag{4}$$

This property of the model is essential to carry out an efficient sampling. Furthermore, one can uniquely determine any σ by the $n-1$ integers $V_1(\sigma)$, $V_2(\sigma)$,..., $V_{n-1}(\sigma)$ defined as

$$V_j(\sigma, I) = \sum_{l>j} 1_{[l \prec_\sigma j]} \tag{5}$$

where I denotes the identity permutation (1, 2,... n) and $l \prec_\sigma j$ means that l precedes j (i.e. is preferred to j) in permutation σ.

3.2 Learning and Sampling a Mallows Model

At each step of the EDA, we need to learn a Mallows model from the set of selected individuals (permutations). Therefore, given a dataset of permutations $\{\sigma_0, \sigma_1, \ldots, \sigma_N\}$ we need to estimate σ_0 and θ. In order to do that, we use the maximum likelihood estimation method. The log-likelihood function can be written as

$$log\ l(\sigma_1, ..., \sigma_N|\sigma_0, \theta) = -N \sum_{j=1}^{n-1} (\theta \bar{V}_j + \log \psi_j(\theta)) \tag{6}$$

where $\bar{V}_j = \sum_{i=1}^{N} V_j(\sigma_i, \sigma_0)/N$, i.e. $\bar{V}_j$ denotes the observed mean for V_j.

The problem of finding the central permutation or consensus ranking is called rank aggregation and it is, in fact, equivalent to finding the MLE estimator of σ_0, which is NP-hard. One can find several methods for solving this problem, both exact [7] and heuristic [13]. In this paper we propose the following: first, the average of the values at each position is calculated, and then, we assign index 1 to the position with the lowest average value, next index 2 to the second lowest position, and so on until all the n values are assigned.

Once σ_0 is known, the estimation of θ maximizing the log-likelihood is immediate by numerically solving the following equation:

$$\sum_{j=1}^{n-1} \bar{V}_j = \frac{n-1}{e^{\theta}-1} - \sum_{j=1}^{n-1} \frac{n-j+1}{e^{(n-j+1)\theta}-1} \tag{7}$$

In general, this solution has no closed form expression, but can be solved numerically by standard iterative algorithms such as Netwon-Rapshon.

In order to sample, we consider a bijection between the V_j-s and the permutations. By sampling the probability distribution of the V_j-s defined by (8), a V_j-s vector is obtained. The new permutations are calculated applying the sampled V_j vector to the consensus permutation σ_0 following a specific algorithm [14].

$$P[V_j(\sigma\sigma_0^{-1}, I) = r] = \frac{e^{-\theta r}}{\psi_j(\theta)} \tag{8}$$

4 Experiments

Once the Mallows model has been introduced, we devote this section to carrying out some experiments in order to analyze the behavior of this new EDA. As stated previously, the variance of the Mallows model is controlled by a spread parameter θ, and therefore it will be necessary to observe how the model behaves according to different values of θ. In a second phase, and based on the values previously obtained, the Mallows EDA will be run for some instances of the FSP problem. In addition, for comparison purposes, two state-of-the-art EDAs [5] will be also included, in particular Tsutsui's EHBSA and NHBSA approaches.

4.1 Analysis of the Spread Parameter θ

As can be seen in the description of the Mallows model, the spread parameter θ will be the key to control the trade-off between exploration and exploitation. As shown in Fig. 1, as the value of θ increases, the probability tends to concentrate on a particular permutation (solution). In order to better analyze this behavior, we have run some experiments, varying the values of θ and observing the probability assigned to the consensus ranking (σ_0). Instances of different sizes (10, 20, 50, and 100) and a wide range of θ values (from 0 to 10) have been studied. The results shown in Fig. 2 demonstrate how, for low values of θ, the probability of σ_0 is quite small, thus encouraging a exploration stage. However, once a threshold is exceeded, the probability assigned to σ_0 increases quickly, leading the algorithm to an exploitation phase.

Based on these results, we completed a second set of experiments executing the Mallows EDA on some FSP instances. The θ parameter was fixed using a

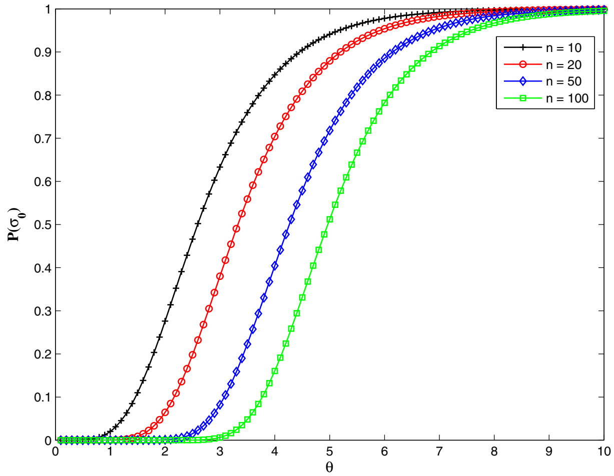

Fig. 2. Probability assigned to σ_0 for different θ and n values

range of promising values extracted from the previous experiment. Particularly, we decided to use 8 values in the range [0,2]. These values are {0.00001, 0.0001, 0.001, 0.01, 0.1, 0.5, 1, 2}. The rest of the parameters typically used in EDAs are presented in Table 1. Regarding the FSP instances, the first instance of each set *tai20×5, tai20×10, tai50×10*, *tai100×10* and *tai100×20*[1] was selected. Each experiment was run 10 times. Table 2 shows the error rate of these executions. This error rate is calculated as the normalized difference between the best value obtained by the algorithm and the best known solution.

Table 1. Execution parameters of the algorithms. Being n the problem size.

Parameter	Value
Population size	$10n$
Selection size	$10n/2$
Offspring size	$10n - 1$
Selection type	Ranking selection method
Elitism selection method	The best individual of the previous generation is guaranteed to survive
Stopping criteria	$100n$ maximum generations or $10n$ maximum generations without improvement

The results shown in 2 indicate that the lowest or highest values of θ (in the [0,2] interval) provide the worst results, and as θ moves inside the interval the performance increases. Particularly, the best results are obtained for 0.1, 0.5 and 1 values.

[1] Éric Taillard's web page. http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html

Table 2. Average error rate of the Mallows EDA with different constant θs

θ	20×5	20×10	50×10	100×10	100×20
0.00001	0.0296	0.0930	0.1359	0.0941	0.1772
0.0001	0.0316	0.0887	0.1342	0.0917	0.1748
0.001	0.0295	0.0982	0.1369	0.0910	0.1765
0.01	0.0297	0.0954	0.1275	0.0776	0.1629
0.1	0.0152	0.0694	0.0847	**0.0353**	**0.1142**
0.5	**0.0081**	0.0347	**0.0780**	0.0408	0.1236
1	0.0125	**0.0333**	0.0936	0.0610	0.1444
2	0.0182	0.0601	0.1192	0.0781	0.1649

4.2 Testing the Mallows EDA on FSP

Finally, we decided to run some preliminary tests for the Mallows EDA algorithm on the previously introduced set of FSP instances (taking in this case the first six instances from each file). Taking into account the results extracted from the analysis of θ, we decided to fix its initial value to 0.001, and to set the upper bound to 1. The parameters described in Table 1 were used for the EDAs. In particular, for NHBSA and EHBSA algorithms, B_{ratio} was set to 0.0002 as suggested by the author in [20].

For each algorithm and problem instance, 10 runs have been completed. In order to analyze the effect of the population size on the Mallows model, in addition to $10n$ we have also tested n, $5n$ and $20n$ sizes.

Table 3 shows the average error and standard deviation of the Mallows EDA and Tsutsui's approaches regarding the best known solutions. Note that each entry in the table is the average of 60 values (6 instances × 10 runs). Looking at these results, it can be seen that Tsutsui's approaches yield better results for small instances. However, as the size of the problem grows, both approaches obtain similar results for 50 × 20 instances, and the Mallows EDA shows a better performance for the biggest instances 100 × 10 and 100 × 20. The results obtained show that the Mallows EDA is better for almost all population sizes. These results stress the potential of this Mallows EDA approach for permutation-based problems.

Table 3. Average error and standard deviation for each type of problem. Results in bold indicate the best average result found.

EDA		Mallows				EHBSA	NHBSA
		n	*5n*	*10n*	*20n*	*10n*	*10n*
20×5	*avg.*	0.0137	0.0102	0.0102	0.0096	**0.0039**	0.0066
	dev.	0.0042	0.0037	0.0035	0.0039	0.0034	0.0032
20×10	*avg.*	0.0357	0.0258	0.0250	0.0232	**0.0065**	0.0076
	dev.	0.0054	0.0033	0.0037	0.0030	0.0023	0.0016
50×10	*avg.*	0.0392	0.0345	0.0342	0.0349	**0.0323**	0.033
	dev.	0.0067	0.0071	0.0059	0.0062	0.0066	0.0069
100×10	*avg.*	0.0093	**0.0078**	0.0083	0.0089	0.0199	0.0157
	dev.	0.0040	0.0040	0.0045	0.0053	0.0047	0.0062
100×20	*avg.*	**0.0583**	0.0610	0.0661	0.0587	0.0676	0.0631
	dev.	0.0116	0.0130	0.0132	0.0121	0.0050	0.0071

5 Conclusions and Future Work

In this paper a specific EDA for dealing with permutation-based problems was presented. We introduced a novel EDA, that unlike previously designed permutation based EDAs, is intended for codifying probabilities over permutations by means of the Mallows model. In order to analyze the behavior of this new proposal, several experiments have been conducted. Firstly, the θ parameter has been analyzed, in an attempt to discover its influence in the exploration-exploitation trade-off. Secondly, the Mallows EDA has been executed over several FSP instances using the information extracted from θ values in the initial experiments. Finally, for comparison purposes, two state-of-the-art EDAs have been executed: EHBSA and NHBSA. From these preliminary results, it can be concluded that the Mallows EDA approach presents an interesting behavior, obtaining better results than Tsutsui's algorithms as the size of the problem increases.

As future work, there are several points that deserve a deeper analysis. On the one hand, it would be interesting to extend the analysis of θ in order to obtain a better understanding of its influence: initial value, upper bound, etc. On the other hand, with the aim of ratifying these initial results it would be interesting to test this Mallows EDA on a wider set of problems, such as the Traveling Salesman Problem, the Quadratic Assignment Problem or the Linear Ordering Problem.

Acknowledgments. We gratefully acknowledge the generous assistance and support of Ekhine Irurozki and Prof. S. Tsutsui in this work. This work has been partially supported by the Saiotek and Research Groups 2007-2012 (IT-242-07) programs (Basque Government), TIN2010-14931 and Consolider Ingenio 2010 - CSD 2007 - 00018 projects (Spanish Ministry of Science and Innovation) and COMBIOMED network in computational biomedicine (Carlos III Health Institute). Josu Ceberio holds a grant from Basque Goverment.

References

1. Bean, J.C.: Genetic Algorithms and Random Keys for Sequencing and Optimization. INFORMS Journal on Computing 6(2), 154–160 (1994)
2. Bengoetxea, E., Larrañaga, P., Bloch, I., Perchant, A., Boeres, C.: Inexact graph matching by means of estimation of distribution algorithms. Pattern Recognition 35(12), 2867–2880 (2002)
3. Bosman, P.A.N., Thierens, D.: Crossing the road to efficient IDEAs for permutation problems. In: Spector, L., et al. (eds.) Proceedings of Genetic and Evolutionary Computation Conference, GECCO 2001, pp. 219–226. Morgan Kaufmann, San Francisco (2001)
4. Bosman, P.A.N., Thierens, D.: Permutation Optimization by Iterated Estimation of Random Keys Marginal Product Factorizations. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 331–340. Springer, Heidelberg (2002)

5. Ceberio, J., Irurozki, E., Mendiburu, A., Lozano, J.A.: A review on Estimation of Distribution Algorithms in Permutation-based Combinatorial Optimization Problems. Progress in Artificial Intelligence (2011)
6. Ceberio, J., Mendiburu, A., Lozano, J.A.: A Preliminary Study on EDAs for Permutation Problems Based on Marginal-based Models. In: Krasnogor, N., Lanzi, P.L. (eds.) GECCO, pp. 609–616. ACM (2011)
7. Cohen, W.W., Schapire, R.E., Singer, Y.: Learning to order things. In: Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems, NIPS 1997, vol. 10, pp. 451–457. MIT Press, Cambridge (1998)
8. Fligner, M.A., Verducci, J.S.: Distance based ranking Models. Journal of the Royal Statistical Society 48(3), 359–369 (1986)
9. Gupta, J., Stafford, J.E.: Flow shop scheduling research after five decades. European Journal of Operational Research (169), 699–711 (2006)
10. Larrañaga, P., Lozano, J.A.: Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers, Dordrecht (2002)
11. Lozano, J.A., Mendiburu, A.: Solving job schedulling with Estimation of Distribution Algorithms. In: Larrañaga, P., Lozano, J.A. (eds.) Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation, pp. 231–242. Kluwer Academic Publishers (2002)
12. Mallows, C.L.: Non-null ranking models. Biometrika 44(1-2), 114–130 (1957)
13. Mandhani, B., Meila, M.: Tractable search for learning exponential models of rankings. In: Artificial Intelligence and Statistics (AISTATS) (April 2009)
14. Meila, M., Phadnis, K., Patterson, A., Bilmes, J.: Consensus ranking under the exponential model. In: 22nd Conference on Uncertainty in Artificial Intelligence (UAI 2007), Vancouver, British Columbia (July 2007)
15. Mühlenbein, H., Paaß, G.: From Recombination of Genes to the Estimation of Distributions I. Binary Parameters. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996, Part IV. LNCS, vol. 1141, pp. 178–187. Springer, Heidelberg (1996)
16. Pelikan, M., Goldberg, D.E.: Genetic Algorithms, Clustering, and the Breaking of Symmetry. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, Springer, Heidelberg (2000)
17. Robles, V., de Miguel, P., Larrañaga, P.: Solving the Traveling Salesman Problem with EDAs. In: Larrañaga, P., Lozano, J.A. (eds.) Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers (2002)
18. Romero, T., Larrañaga, P.: Triangulation of Bayesian networks with recursive Estimation of Distribution Algorithms. Int. J. Approx. Reasoning 50(3), 472–484 (2009)
19. Tsutsui, S.: Probabilistic Model-Building Genetic Algorithms in Permutation Representation Domain Using Edge Histogram. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 224–233. Springer, Heidelberg (2002)
20. Tsutsui, S., Pelikan, M., Goldberg, D.E.: Node Histogram vs. Edge Histogram: A Comparison of PMBGAs in Permutation Domains. Technical report, Medal (2006)