



# Article Multiobjective Learning to Rank Based on the (1 + 1) Evolutionary Strategy: An Evaluation of Three Novel Pareto Optimal Methods

Walaa N. Ismail <sup>1,2</sup>, Osman Ali Sadek Ibrahim <sup>3</sup>, Hessah A. Alsalamah <sup>4,5,\*</sup> and Ebtesam Mohamed <sup>2</sup>

- <sup>1</sup> Department of Management Information Systems, College of Business, Al Yamamah University, Riyadh 11512, Saudi Arabia
- <sup>2</sup> Faculty of Computers and Information, Minia University, Minia 61519, Egypt; ebtesam.mohamed@mu.edu.eg
   <sup>3</sup> Department of Computer Science, Minia University, Minia 61519, Egypt
  - <sup>4</sup> Information Systems Department, College of Computer and Information Sciences, King Saud University, Rivadh 4545, Saudi Arabia
  - <sup>5</sup> Department of Computer Engineering, College of Engineering and Architecture, Al Yamamah University, Riyadh 11512, Saudi Arabia
  - \* Correspondence: halsalamah@ksu.edu.sa

Abstract: In this research, the authors combine multiobjective evaluation metrics in the (1 + 1) evolutionary strategy with three novel methods of the Pareto optimal procedure to address the learning-to-rank (LTR) problem. From the results obtained, the Cauchy distribution as a random number generator for mutation step sizes outperformed the other distributions used. The aim of using the chosen Pareto optimal methods was to determine which method can give a better exploration–exploitation trade-off for the solution space to obtain the optimal or near-optimal solution. The best combination for that in terms of winning rate is the Cauchy distribution for mutation step sizes with method 3 of the Pareto optimal procedure. Moreover, different random number generators were evaluated and analyzed versus datasets in terms of NDCG@10 for testing data. It was found that the Levy generator is the best for both the MSLR and the MQ2007 datasets, while the Gaussian generator is the best for the MQ2008 dataset. Thus, random number generators clearly affect the performance of ES-Rank based on the dataset used. Furthermore, method 3 had the highest NDCG@10 for MQ2008 and MQ2007, while for the MSLR dataset, the highest NDCG@10 was achieved by method 2. Along with this paper, we provide a Java archive for reproducible research.

**Keywords:** evolutionary strategy; learning to rank; LETOR; Microsoft Bing; multiobjective optimization; Pareto optimal

## 1. Introduction

In information retrieval (IR), ranking retrieved documents according to their relevance to a user query is an important task. To adjust the relevance of the retrieved documents, a ranking system needs to be used after receiving the user's query, as shown in Figure 1. An optimization model is used to order the collection of available documents using such a ranking system [1,2]. A number of unsupervised term vector models (TVMs), including the vector space model (VSM), TF-IDF and Okapi BM25, were used in early IR research [2,3]. Based on these models, the documents that were retrieved were rated in terms of their relevance to the user's search terms following one scoring method (TWS) in IR systems. It was found that these methods were insufficient for the development of effective IR systems. There are several reasons for this, including the fact that scoring approaches such as Okapi BM25 and various language models are limited in their ability to return appropriate search results based on relevance judgments [3,4]. Consequently, multiple scoring methods should be used to rank retrieved documents based on the user's query. Furthermore, other aspects, such as the importance of business documents on the web,



Citation: Ismail, W.N.; Ibrahim, O.A.S.; Alsalamah, H.A.; Mohamed, E. Multiobjective Learning to Rank Based on the (1 + 1) Evolutionary Strategy: An Evaluation of Three Novel Pareto Optimal Methods. *Electronics* **2023**, 12, 3724. https://doi.org/10.3390/ electronics12173724

Academic Editors: Alireza Mousavi and Zhengwen Huang

Received: 3 August 2023 Revised: 25 August 2023 Accepted: 29 August 2023 Published: 4 September 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). should also be considered. Among other desirable features, the host server is taken into account when ranking documents. A statistical machine learning approach traditionally focuses on solving one single-objective optimization problem [4,5], that is, during a training set, it is necessary to minimize the average loss. Several additional quantities, including the complexity of the model, are either implicitly addressed by the choice of the model class or are incorporated into the main objective by incorporating weighted regularization terms.



Figure 1. Learning-to-rank process view according to [6].

Recently, the machine learning community has focused on additional quantities of interest, such as the fairness, robustness, efficiency or interpretability of learned models. Optimizing these can conflict with the goal of reducing training loss, and task-specific trade-offs need to be considered. The problem with hard-coding such trade-offs is that they may have undesirable consequences. The process of selecting them is also cumbersome when multiple objectives are at stake. There has been an increase in interest in multiobjective learning in recent years as a way to avoid the need for a priori trade-offs. By performing multiobjective optimization at the same time as training the actual model, the optimization either finds promising trade-off parameters simultaneously or computes multiple solutions that reflect different trade-offs, ideally along the Pareto frontier. Despite being algorithmically rich, the theory of multiobjective optimization and learning has been little studied. Specifically, learning theory results such as generalization bounds are almost completely absent. To overcome the mentioned limitations, a new approach is proposed in this study that involves combining multiobjective evaluation metrics in (1 + 1)—an evolutionary strategy—with three different methods and examining their effectiveness with a single-objective evolutionary strategy. The contributions can be summarized as follows:

- A hybrid multiobjective algorithm is proposed for a more accurate exploration of the IR problem search space. This objective is achieved by devising the multiobjective evolutionary strategy with three different methods.
- The performance of the multiobjective evolutionary strategy is enhanced by automatically choosing and optimizing search results by using three novel multiobjective functions to determine which set of solutions are nondominant with regard to one another and are superior to the rest in the search space.
- A comprehensive experiment was conducted to validate the effectiveness of the proposed strategy and to compare its performance against that of state-of-the-art singleobjective evolutionary algorithms.

 Detailed code for all experiments has been posted at https://www.researchgate.net/ publication/364265942\_Multi-Objective\_11-Evolutionary\_Strategy\_Multi-ESRank\_ for\_Learning\_to\_Ranking\_Problem, accessed on 20 October 2022. This provides the first example of multiobjective learning-to-rank Java archive applications of the method proposed for research reproducibility.

## 2. Related Work

In this section, we introduce/discuss some related studies that applied multiobjective methods to LTR problems.

A learning-to-rank process aims to produce a ranking model that is capable of predicting accurately the relevance of a set of queries and items, improving user satisfaction and engagement. To obtain a ranking function, a structured process involving several steps is required. Firstly, a dataset is gathered which includes queries, items and relevance labels, ensuring a variety of scenarios for robustness. Following this, relevant features are extracted from both queries and items, capturing the critical aspects that affect the relative relevance of both. Once the training data have been obtained, they data are used to develop a ranking function. Finally, a ranking list of documents associated with a new query is created by using the ranking function [7,8].

The study in [8] presented a multiobjective LTR approach for commercial search engines using LambdaMART, a state-of-the-art ranking algorithm. They modified the  $\lambda$  functions to solve two associated problems with the current LambdaMART  $\lambda$ -gradient. The goal was to stop the ranking model from trying to separate documents that were already ordered and separated in addition to making ranking mistakes that persisted long into training. Their proposed approach achieved significant improvements in terms of accuracy over the baseline state-of-the-art ranker LambdaMART. The experiments were performed on a large real-world dataset in which each query–URL pair had 860 features. However, the dataset itself and the authors' code package are not available to researchers for research reproducibility.

The incorporation of relevant and well-engineered features into the dataset will enhance the model's ability to generalize and provide informed ranking results. Several evolutionary multiobjective feature-selection ranking algorithms have been proposed in recent years [7,8]. According to Li et al. [9], a new decomposition-based multiobjective immune algorithm called MOIA/DFSRank was proposed for the selection of features in L2R. To ensure greater convergence and diversity of the initial populations, representative features are selected for generation based on their importance and redundancy score. The proposed algorithm utilizes two effective operators: clonal selection and mutation, where the clonal selection operator generates clones to facilitate the search direction during evolution, while the mutation operator retains excellent features with a high probability of evolution. Kundu, P.P. et al. [10] employed the NSGA-II algorithm framework to introduce a method for feature selection utilizing an SNN-based distance metric. This method aims to concurrently maximize both the count of selected features and the classification accuracy. Zhang et al. [11] utilized an enhanced MOPSO algorithm to effectively diminish the Hamming loss value, even when utilizing a reduced number of features. In a related context, Das, A. [12] presented a multiobjective evolutionary algorithm centered on relevance and redundancy considerations. This approach demonstrated superior classification outcomes while utilizing a reduced set of selected features. Mahapatr et al. [13] addressed the multiobjective optimization (MOO) problem associated with multilabel LTR (training a model using a different relevance criterion). Essentially, this framework is capable of consuming any first-order gradient-based MOO algorithm to train a ranking model. Cheng et al. [14], on the other hand, addressed the learning-to-rank problem by devising an algorithm grounded in the NSGA-II framework, yielding commendable results. Nevertheless, there remains a need for further enhancement of classification accuracy within this framework.

For commercial search engine preferences, the query–item relevance can be judged based on different criteria. For instance, in a search for products, the search engine may rank products based on their quality or on the user's price preferences. The research study in [4] applied several multiobjective optimization methods with preference directions, such as the traditional Pareto optimal search, to LTR problems. Their approach was applied to three LTR datasets and worked effectively for all three datasets. The datasets included the Microsoft Learning-to-Rank web search dataset (MSLR-WEB30K) [15], which is represented by a 136-dimensional feature vector, and E-commerce datasets. They presented the maximum weighted loss as a novel model evaluation metric. The gradient-boosted regression tree (GBRT or MART) [16] algorithm was used in the study. They found that the single-objective MART outperformed the multiobjective MART. Thus, they proposed a smooth remedy procedure to improve the performance of multiobjective MART compared to using the traditional Pareto optimal method in this algorithm.

Multiobjective optimization methods have been developed and used for multitask learning, especially for combinatorial optimization; however, their applications to LTR problems are still a novel research topic.

A different line of research studies presented a multiobjective learning framework where the authors used relevance labels and adjusted remedies for the ranking function to satisfy multiple objectives to produce results satisfying specific criteria, such as scale calibration [17] and fairness [18,19]. They used the Rank Neural Network (RankNET), LambdaMART and Listwise Neural Network (ListNET) approaches [16]. Remedy procedures were used to overcome the gaps in performance between the single-objective approaches and multiobjective ones [15]. On the other hand, evolutionary strategy LTR (ES-Rank) outperformed MART, RankNET, LambdaMART and ListNET in previous research [16]. Furthermore, the single-objective ES-Rank outperformed the 14 well-known evolutionary and machine learning approaches.

Hence, the principal objective of this research is to introduce an innovative searchspace-exploration-procedures-based multiobjective algorithm using the Pareto optimal approach. These procedures are used here as a remedy between single- and multiobjective performance for the same algorithm to prove that the multiobjective version of the LTR algorithm can outperform the single-objective version in some exploration circumstances. Empirical findings attest to the heightened performance of the introduced algorithm in tackling the challenges posed by the learning-to-rank problem.

### 3. Proposed Approach

In the field of optimization, metaheuristic algorithms are computational techniques for solving complex optimization problems. Traditionally, optimization methods may struggle when faced with this type of problem because of its size or nonlinearity or the presence of multiple objectives that conflict. A metaheuristic is an approach to optimization that is different from an exact optimization algorithm. While exact algorithms promise the best solution given enough time and resources, metaheuristics offer approximate solutions that are often of excellent quality. A single-objective heuristic addresses optimization problems based on a single objective. The objective of these problems is to maximize or minimize a single criterion or goal. By utilizing heuristics, an optimal solution to the given objective function can be found. Multiobjective heuristics, on the other hand, are designed to solve optimization problems with multiple conflicting objectives. This paper uses the (1 + 1) evolutionary strategy algorithm for learning to rank (ES-Rank) in two variations, which are single-objective and multiobjective evaluation metrics (as shown in Figure 2).

The single-objective ES-Rank has been used in the previous study in [16] in comparison with 14 evolutionary and machine learning methods, and it outperformed them. It is often necessary to optimize multiple criteria simultaneously in such problems, and these objectives often conflict. In general, it is not possible to find a solution that optimizes all objectives simultaneously due to inherent trade-offs. Several studies have shown that multiobjective optimization is usually less accurate than the approach of optimizing each fitness function individually. However, our method can be a strong rival to the single-objective ES-Rank.



Figure 2. Flowchart representation of the proposed algorithm.

This study aims to find the most effective method for multiobjective learning in order to optimize performance in multiobjective learning. The present study introduces three methods, two of which are novel methods in the field of multiobjective optimization.

Our proposed optimization algorithm, the multiobjective (1 + 1) evolutionary strategy, is a novel approach for tackling complex multi-ES-Rank problems. The problem involves multiple objectives to be optimized, and no single solution may be considered to be the best across all objectives. In this algorithm, the decision variables are assigned random values for a population of "individuals", each representing a potential solution. In order to rank these individuals, this algorithm employs the Pareto principle. In a multiobjective optimization problem, the Pareto optimal, also known as the Pareto frontier, is a set of solutions that are not considered to be dominated by any of the objectives. As a result, no solution in the set is superior to any of its competitors in all of its objectives, and at least one objective has improved without compromising any of the others. The framework of the proposed multiobjective (1 + 1) evolutionary strategy is as follows.

#### 3.1. Step 1: Initialization

Set initial values for the maximum algorithm iterations and population size, and then generate an initial population of candidate solutions (individuals), denoted as P(0), and assign random values to the decision variables for each individual.

## 3.2. Step 2: Termination

If the termination condition has not been reached for a maximum number of iterations, then continue; otherwise, print out the Pareto optimal set from P.

## 3.3. Step 3: Mutation

Using the objective function values for each individual, calculate the fitness value for that individual. In order to calculate the fitness value, a ranking-based approach can be used, such as a nondominated sorting rank. We utilized 3 different methods from a single fitness objective function, ES-Rank. These 3 multiobjective ES-Rank methods use the Pareto

frontier approach for the cumulative objective function *MFitness*. This cumulative fitness function can be calculated by Equation (1).

$$MFittness = \sum 5i = 1C_i \times Fittness_i \tag{1}$$

where  $C_i$  is the Pareto frontier coefficient *i*, which corresponds to the fitness evaluation metric *Fittness<sub>i</sub>*, and *i* is an integer number between 1 and 5. The fitness evaluation metrics used in this study are the mean average precision (MAP), normalized discounted cumulative gain (NDCG@10), reciprocal rank (RR@10), expected reciprocal rank (ERR@10) and precision (P@10) at top 10 documents retrieved [20]. The 3 multiobjective ES-Rank methods use 3 different representations for  $C_i$ :

- The first multiobjective ES-Rank approach uses  $C_i = 11 \forall i$ , while  $i = \{1, 2, 3, 4, 5\}$ .
- The second multiobjective ES-Rank approach uses a traditional real random number generator for assigning a real number value for the  $C_i$  coefficient for every fitness function in every evolving iteration with constraints. This constraint is that  $\sum 5i = 1C_i = 1$  in every evolving iteration.
- The third multiobjective ES-Rank approach uses a ziggurat Gaussian random number generator to assign a real number value for the  $C_i$  coefficient for every fitness function in every evolving iteration with constraints. This constraint is that  $\sum_{i=1}^{5} C_i$  in every evolving iteration. The ziggurat Gaussian random number generator [21] generates a normalized Gaussian random number between 0 and 1 rather than -50 and 50 as in the traditional Gaussian random number generator.

### 3.4. Step 4: Population Evolution

To guarantee that the constraints in the second and third multiobjective ES-Rank on Pareto frontier coefficients are met, we assume that the five Pareto frontier coefficients generated using random number generators are  $C_i = \{C_1, C_2, C_3, C_4, C_5\}$  in each evolving iteration. There is no guarantee for the summation value for these coefficients to be 1 without a normalization factor. The normalization factor *Nfactor* can be calculated by Equation (2).

$$Nfactor = \frac{1}{(C_1 + C_2 + C_3 + C_4 + C_5)}$$
(2)

Then, the Pareto coefficients are calculated by  $C_i = Nfactor \times C_i$ , where  $i \le 5$ .

During each iteration, methods 2 and 3 use multiobjective randomization functions based on traditional and ziggurat Gaussian distribution Pareto coefficients. In this manner, more exploration can be achieved for a multiobjective search-space solution, while exploitation can be limited to a single Pareto coefficient sum. A rank (r) is assigned to each individual, with a lower rank indicating a higher level of fitness. Ranks and fitness values are then used to select parents for reproduction. The probability of becoming a parent increases for individuals with a lower rank and a higher fitness value.

### 3.5. Step 5: Population Update

Evolutionary strategy consists of two solutions, the current solution (parent) and a candidate solution (offspring) that results from perturbing the parent. If offspring are not at least as efficient as their parents, they will be discarded from consideration for the following generation. As a vector of weights, the chromosome represents the evolving ranking model.

Algorithm 1 outlines the multi-ES-Rank algorithm. The training and validating set of query–document pairs provides a means of assessing evolutionary solutions in each iteration, and the output of this algorithm is a ranking model for the dataset used in the evolving phase. Using PCh as a parent chromosome, each gene is represented as a real number, representing the significance of the corresponding feature for ranking the training and validating data instances, where the data instances are queries and documents. Each gene in steps 1 through 4 is initialized to a value of 0.5 in the parent chromosome vector. The Boolean parameter Good is used to indicate whether to repeat the previous mutation steps from the previous generation or not. It is set to FALSE in step 5 when the previous mutation steps are to be repeated.

A copy of PCh is assigned to OffCh in step 6. The evolving process is repeated until the maximum generation MaxGenerations is reached; the number of iterations is 1300 in this paper. The evolving procedure begins in step 7 and ends in step 24. The procedure for managing mutations is demonstrated in steps 8–16 by choosing the number of genes to mutate (RM). Four probability distributions are used to determine the mutation step (steps 11 to 15): Gaussian, Cauchy, Levy and uniform. The successful evolution process (which produced good offspring) for evolving iteration G is repeated in evolving iteration G, as illustrated in steps 11 to 15. Using the fitness metrics, steps 17 to 23 determine which PCh or OffCh to use. Finally, in step 25, the relationship between dynamic feature weights and query–document pairs is represented by the mathematical transposition of the feature weights vector (i.e., multi-ES-Rank procedure).

### Algorithm 1: MultiES-Rank: Multiobjective Evolutionary Strategy Ranking Approach

*Input:* A training set  $\alpha(q, d)$  and a validation set  $\eta(q, d)$  of query-document pairs of feature vectors. *Output:* A linear ranking function F(q, d) that assigns a weight to every query-document pair indicating its relevancy degree.

1 Initialization: 2 For (Gen\_i ∈ PCh) do 3  $Gen_i = 0.0;$ 4 end 5 Good = FALSE; OffCh = PCh;6 For (G = 1 to MaxGenerations) do 7 8 If (Good==TRUE) Then Use the same mutation process of generation (G-1) on OffCh to mutate next OffCh, that is, mutate the 9 same RM genes using the same Mutation Step; 10 Else 11 Choose number of genes to mutate RM at random from 1 to M 12 For (j = 1 to RM)13 Choose random Gen\_i in OffCh for mutation; 14 Mutate Gene\_i using Mutation Step according to Probability Distributions used 15 end end 16 17 If (((Fitness(PCh,  $\alpha(q,d))$  < Fitness(OffCh  $\alpha(q,d)$ )) & (Fitness(PCh,  $\eta(q,d)) \leq$  Fitness(OffCh,  $\eta(q,d)$ ))) Then 18 PCh = OffCh;19 Good=TRUE; 20 Else 21 OffCh = PCh;22 Good = FALSE; 23 end **Return**: The linear ranking function F(q, d) = PCh, that is PCh at the end of the MaxGenerations contains the evolved vector W of M 24

## feature weights.

#### 4. Experimental Results

This section includes a thorough experimental investigation that compares the three proposed learning-to-rank multiobjective strategies to a single-objective existing approach in terms of five accuracy fitness metrics. MAP, RR, ERR, NDCG and P (mean average precision, reciprocal rank, expected reciprocal rank—total precision, normalized discount cumulative gain and average precision) at top 10 documents retrieved are the five metrics used to evaluate accuracy, as stated in subsection IV-A. To evaluate the performance of an LTR approach, the LTR technique is first applied to the training set. Afterwards, the ranking model's performance is evaluated using the test set to determine how well the LTR algorithm makes predictions.

#### 4.1. Benchmark Datasets and Evaluation Fitness Metrics

Three benchmarking datasets are considered in this paper, as follows:

The MSLR-WEB30K dataset [22]: This dataset provides a comprehensive and realistic set of query–document pairs with relevance labels. Additionally, there is a set of features associated with each query–document pair that capture various aspects of the query and the document. Among these features are textual features, numerical features and other metadata that can be used to determine the degree of relevance of a document with respect to a specific query.

LETOR 4.0 [23,24]: It is part of the LETOR (Learning to Rank for Information Retrieval) dataset collection. A significant number of query–document pairs are included in the dataset, each associated with a relevance label. Additionally, a variety of features are included in the dataset that capture the characteristics of both queries and documents. These include textual attributes, numerical attributes and other metadata. These features are designed to aid ranking algorithms in determining the relevance of documents to a query.

As can be seen in Table 1, these datasets have a number of different characteristics. Compared to LETOR 4 datasets (MQ2007 and MQ2008), the Microsoft Bing Search dataset (MSLR-WEB30K) has a much higher number of query-document pairs and features. There are several low-level characteristics associated with each query-document pair, such as term frequency and inverse document frequency. In order to determine low-level features for all document parts (title, anchor, body and whole), a set of low-level features was determined. In addition, there are high-level features that indicate how well the searches and documents correspond. Additionally, hybrid features have been employed in previous SIGIR conference papers including LMIR.ABS, LMIR.JM, LMIR.DIR and LMIR.DIR as well as the Language Model with Absolute Discounted Smoothing [22–25] and Language Model with Jelinek–Mercer smoothing [LMIR.JM]. There are 30,000 queries in the MSLR-WEB30K dataset. MQ2008 contains fewer than 1000 queries, whereas MQ2007 contains 1692 queries. There are a variety of query-document combinations for each query, which are based on a set of relevant and irrelevant documents. A relevance label indicates the level of relevance of a query when it is accompanied by a document (relationship query–document). As a general rule, relevance labels are classified as 0 (for totally irrelevant), 1 (for moderately relevant) and 2 (for very relevant). There is one exception to this rule, the MSLR-WEB30K dataset, where values range from 0 (irrelevant) to 4 (perfectly relevant).

Dataset	Queries	Query–Document Pairs	Features	Relevance Labels	No. of Folds
MQ2007	1692	69,623	46	{0, 1, 2}	5
MQ2008	784	15,211	46	{0, 1, 2}	5
MSLR-WEB30K	30,000	3, 771, 125	136	$\{0, 1, 2, 3, 4\}$	5

Table 1. Properties of the benchmark datasets used in the experimental study.

In this research, MAP, NDCG@10, P@10, RR@10 and ERR@10 were used as five distinct fitness functions on the training sets [1]. They were also used as assessment measures for the test-set ranking algorithms. These fitness functions were demonstrated in detail in [20].

## 4.2. Result Analysis and Discussion

This section gives an overview of the progress achieved using multiobjective LTR. From the results obtained, we can say that using the Cauchy probability distribution as a random number generator for mutation step sizes in multiobjective ES-Rank outperformed Gaussian, Levy and uniform distributions. It also outperformed single-objective ES-Rank, but the multiobjective method that uses the Cauchy distribution as the dominant method in performance relies on the particular dataset used. Figures 3–5 illustrate the superiority of the proposed methodologies for LTR for the three datasets used.



Figure 3. Winning rate for single-objective vs. multiobjective ES-Rank (MSLR-WEB10K).



Figure 4. Winning rate for single-objective vs. multiobjective ES-Rank (MQ2008).



Figure 5. Winning rate for single-objective vs. multiobjective ES-Rank (MQ2007).

From Figures 3 and 4, for both the MSLR-WEB10K and MQ2008 datasets, the singleobjective ES-Rank performance is higher than the multiobjective ES-Rank. This degradation in performance is accepted in order to gain the multiobjective FS-Rank. This degradation is found that for the dataset MQ2007, the multiobjective ES-Rank with method 1 using uniform and multiobjective ES-Rank with method 2 using Cauchy as a random number generator for mutation step sizes both achieve high performance, with 6 and 7 winning rates, respectively. This is better than the overall performance of the single-objective ES-Rank. These results ensure the effectiveness of our proposed methods for both singleobjective and multiobjective optimization. Moreover, the dataset affects the performance of ES-Rank for all the methods used.

To evaluate the methods of generated random numbers distribution, Figure 6 illustrates the NCDG@10 for the test set MSLR dataset. From Figure 6, we can conclude that Levy is the best one for single-objective ES-Rank, while for multiobjective, Figure 6 shows grouping results based on the method of optimization, where Levy is the best for method 1, method 2 and method 3. Thus, Levy probability distribution as a random number generator for mutation step sizes is recommended for single-objective and multiobjective ES-Rank using all three methods. Moreover, method 2 with Levy achieves the highest NDCG@10 for the MSLR dataset.



Figure 6. NDCG@10 relevance range for testing data (MSLR dataset) grouped by method.

For analyzing and evaluating different random number generators, Figures 6–8 illustrate the NDCG@10 for testing data for MSLR, MQ2007 and MQ2008. For the MQ2008 dataset, Figure 7 illustrates the NCDG@10 for the test set. From Figure 7, it is found that the Gaussian probability distribution as a random number generator for mutation step sizes is recommended for single-objective and multiobjective ES-Rank using all three methods. Moreover, method 3 with Gaussian achieves the highest NDCG@10 for the MQ2008 dataset.

For the MQ2007 dataset, Figure 8 illustrates the NCDG@10 for the test set. From Figure 8, it is found that Levy probability distribution as a random number generator for mutation step sizes is recommended for multiobjective ES-Rank using all three methods; however, Gaussian is recommended for single-objective ES-Rank. Moreover, method 3 with Levy achieves the highest NDCG@10 for the MQ2008 dataset. Thus, random number generators clearly affect the performance of ES-Rank based on the dataset used.



Figure 7. NDCG@10 relevance range for testing data (MQ2008 dataset) grouped by method.



Figure 8. NDCG@10 relevance range for testing data (MQ2007 dataset) grouped by method.

Multi-ES-Rank is an evolutionary strategy that uses a cumulative fitness function to determine the quality of each evolving ranking model in each iteration. Additionally, as the Pareto frontier contains no dominant solution, there is no other solution that performs better on all objectives at the same time. The developed strategy explores the search space and produces diverse solutions reflecting different trade-offs between multiple objectives through the cumulative fitness function. As a result, developed algorithms provide decision-makers with a variety of options from which to select so that they can make informed decisions based on their individual preferences.

In summary, this paper introduces a multiobjective evolutionary strategy (multi-ES-Rank) approach for learning-to-rank problems. In addition, we propose three novel Pareto optimal methods in continuous optimization research. Furthermore, we provide the Java archive package of the proposed approach for research reproducibility. From the experimental results, multi-ES-Rank can outperform single-objective ES-Rank in some circumstances of mutation step sizes and Pareto optimal methods for LTR data, as given in Appendix A. The best performance can be gained with the method using Cauchy as a random number generator for mutation step sizes in terms of winning rate. This causes the multi-ES-Rank to outperform the single-objective ES-Rank in certain conditions. Moreover, the different random number generators are evaluated and analyzed versus the three datasets in terms of NDCG@10 for testing data. It was found that the Levy generator is the best for both the MSLR and MQ2007 datasets while the Gaussian generator is the best for the MQ2008 dataset. Thus, random number generators clearly affect the performance of ES-Rank based on the dataset used. Furthermore, method 3 achieved the highest NDCG@10 for MQ2008 and MQ2007, while for the MSLR dataset, the highest NDCG@10 was achieved by method 2.

An important limitation of this study is the sensitivity of the evolutionary fitness function to configuration parameters. The results of this study highlight the importance of careful parameter tuning, but they also demonstrate that it is difficult to identify a universally optimal configuration because it is often dependent upon specific datasets and problem domains. Since there may not be one configuration suitable for different LTR tasks and datasets, developing automated hyperparameter optimization techniques may mitigate this limitation in the future. This study is also limited by the lack of dedicated multiobjective optimization packages for comparison. Most research focuses on learning-to-rank models with single objectives, such as mean squared error or pairwise ranking. However, in realworld applications, it is often necessary to optimize conflicting objectives simultaneously. The proposed techniques can be further evaluated in more complex optimization scenarios where the performance of the proposed techniques can be evaluated on a broader scale in future research.

## 5. Conclusions and Future Work

In this paper, we describe a general framework for learning to rank using the multiobjective (1 + 1) evolutionary strategy, which can be used with any type of data. As a multiobjective method, the multi-ES-Rank algorithm is based on novel methodologies for calculating cumulative fitness functions. A principled approach to maintaining the relative quality of rankings based on different relevance criteria is provided. Three types of trade-off (fitness calculation) specifications are formalized. The framework was validated using three public datasets, and the source code package is available for reproducible research. A number of directions will be explored to improve the current multi-ES-Rank algorithm in the future, including applying the Pareto optimal methods on some other metaheuristic methods and on some other optimization research domains, and additionally, enhancing the developed package by combining offline and online learning components. The process of offline optimization typically involves the use of historical data in order to train and refine the ranking model, whereas the process of online optimization involves continuous adaptation of the ranking model as a result of real-time user interactions. Using a hybrid approach can enhance the performance and relevance of multi-ES-Rank in dynamic environments by combining the stability and quality of offline optimization with the real-time adaptation of online optimization.

**Author Contributions:** All authors wrote and reviewed the paper (they contributed equally). All authors have read and agreed to the published version of the manuscript.

**Funding:** This research project was supported by a grant from the "Research Center of College of Computer and Information Sciences", Deanship of Scientific Research, King Saud University.

**Data Availability Statement:** Publicly available datasets were analyzed in this study. This data can be found here: [https://www.microsoft.com/en-us/research/project/mslr/].

Conflicts of Interest: The authors declare no conflict of interest.

# Appendix A

MSLR-WEB30K							
	Methods vs. Dataset	MAP	NDCG@10	RR@10	ERR@10	P@10	Winning Rate
	ES-Rank Gaussian	0.5659	0.3766	0.7544	0.2587	0.5879	3
	Multi-ES-Rank 1 Gaussian	0.5386	0.3356	0.7317	0.2623	0.5536	0
	Multi-ES-Rank 2 Gaussian	0.5594	0.3547	0.7541	0.2711	0.5813	0
	Multi-ES-Rank 3 Gaussian	0.5548	0.3622	0.7588	0.2834	0.5815	2
	ES-Rank Cauchy	0.4953	0.2441	0.5865	0.1833	0.4592	1
	Multi-ES-Rank 1 Cauchy	0.4839	0.2457	0.6006	0.1874	0.4598	4
	Multi-ES-Rank 2 Cauchy	0.4678	0.2251	0.5578	0.1687	0.4321	0
Training	Multi-ES-Rank 3 Cauchy	0.4758	0.2408	0.5949	0.1869	0.4492	0
manning	ES-Rank Levy	0.5827	0.3837	0.8020	0.3151	0.6079	3
	Multi-ES-Rank 1 Levy	0.5761	0.3948	0.7982	0.3095	0.6098	0
	Multi-ES-Rank 2 Levy	0.5734	0.4016	0.7992	0.3102	0.6144	2
	Multi-ES-Rank 3 Levy	0.5516	0.3507	0.7602	0.2722	0.5745	0
	ES-Rank uniform	0.4818	0.2357	0.5934	0.1906	0.4624	4
	Multi-ES-Rank 1 uniform	0.4659	0.2316	0.5690	0.1836	0.4278	0
	Multi-ES-Rank 2 uniform	0.4701	0.2336	0.5824	0.1805	0.4383	0
	Multi-ES-Rank 3 uniform	0.4749	0.2397	0.5893	0.1862	0.4463	1
	ES-Rank Gaussian	0.5763	0.3784	0.7657	0.2644	0.5970	3
	Multi-ES-Rank 1 Gaussian	0.5461	0.3415	0.7364	0.2721	0.5608	0
	Multi-ES-Rank 2 Gaussian	0.5671	0.3851	0.7565	0.2789	0.5908	1
	Multi-ES-Rank 3 Gaussian	0.5624	0.3653	0.7613	0.2915	0.5860	1
	ES-Rank Cauchy	0.5041	0.2490	0.5929	0.1896	0.4675	2
	Multi-ES-Rank 1 Cauchy	0.4916	0.2507	0.6074	0.1941	0.4653	3
	Multi-ES-Rank 2 Cauchy	0.4743	0.2276	0.5807	0.1752	0.4343	0
Validation	Multi-ES-Rank 3 Cauchy	0.4832	0.2441	0.6025	0.1929	0.4529	0
validation	ES-Rank Levy	0.5913	0.3911	0.8099	0.3252	0.6188	4
	Multi-ES-Rank 1 Levy	0.5835	0.4003	0.8078	0.3161	0.6186	0
	Multi-ES-Rank 2 Levy	0.5780	0.4021	0.8070	0.3158	0.6168	1
	Multi-ES-Rank 3 Levy	0.5604	0.3579	0.7761	0.2815	0.5831	0
	ES-Rank uniform	0.4910	0.2389	0.5962	0.1976	0.4699	4
	Multi-ES-Rank 1 uniform	0.4722	0.2339	0.5757	0.1882	0.4287	0
	Multi-ES-Rank 2 uniform	0.4760	0.2352	0.5842	0.1859	0.4401	0
	Multi-ES-Rank 3 uniform	0.4819	0.2427	0.5909	0.1913	0.4491	1
	ES-Rank Gaussian	0.5685	0.3710	0.7540	0.2569	0.5879	3
	Multi-ES-Rank 1 Gaussian	0.5404	0.3316	0.7317	0.2623	0.5559	0
	Multi-ES-Rank 2 Gaussian	0.5605	0.3486	0.7541	0.2711	0.5849	0
	Multi-ES-Rank 3 Gaussian	0.5576	0.1581	0.7588	0.2834	0.5874	2
	ES-Rank Cauchy	0.4971	0.2455	0.5709	0.1875	0.4662	2
	Multi-ES-Rank 1 Cauchy	0.4871	0.2457	0.6006	0.1874	0.4665	3
	Multi-ES-Rank 2 Cauchy	0.4696	0.2228	0.5778	0.1687	0.4354	0
Testing	Multi-ES-Rank 3 Cauchy	0.4789	0.2396	0.5949	0.1869	0.4533	0
lesting	ES-Rank Levy	0.5819	0.3789	0.7973	0.3162	0.6071	2
	Multi-ES-Rank 1 Levy	0.5775	0.3902	0.7982	0.3095	0.6152	1
	Multi-ES-Rank 2 Levy	0.5734	0.3957	0.7992	0.3102	0.6132	2
	Multi-ES-Rank 3 Levy	0.557	0.3492	0.7602	0.2722	0.5815	0
	ES-Rank uniform	0.4856	0.2390	0.5906	0.1944	0.4664	4
	Multi-ES-Rank 1 uniform	0.4673	0.2311	0.5921	0.1836	0.4332	1
	Multi-ES-Rank 2 uniform	0.4717	0.2315	0.5824	0.1805	0.4414	0
	Multi-ES-Rank 3 uniform	0.4769	0.2366	0.5893	0.1862	0.4487	0

 Table A1. Average results of the proposed methods using MSLR-WEB30K.

MSLR-WEB30K								
Met	hod Used/Winning Ev	aluation Rate		Winning Rate				
		Caussian	Evolving	6				
		Gaussian	Predictive	3				
		Cauchy	Evolving	3				
Single-objective ES-Rank		educity	Predictive	2				
Shighe objective 25 failur		Levv	Evolving	7				
			Predictive	2				
		Uniform	Evolving	8				
			Predictive	4				
		Coursian	Evolving	0				
	Method 1	Gaussian	Predictive	0				
		Cauchy	Evolving	7				
		Caucity	Predictive	3				
		Levy	Evolving	0				
			Predictive	1				
		Uniform	Evolving	0				
			Predictive	1				
		Courseion	Evolving	1				
		Gaussian	Predictive	0				
		Cauchy	Evolving	0				
Multiobjective ES-Rank			Predictive	0				
	Method 2	Loww	Evolving	3				
		Levy	Predictive	2				
		Uniform	Evolving	0				
		UIIIUIII	Predictive	0				
		Caussian	Evolving	3				
		Gaussian	Predictive	2				
		Cauchy	Evolving	0				
	Mathad 2	Caucity	Predictive	0				
	Method 5	Levin	Evolving	0				
		Levy	Predictive	0				
		Uniform	Evolving	2				
		Uniform	Predictive	0				

 Table A2. The performance of the proposed methods using MSLR-WEB30K.

 Table A3. Average results of the proposed methods using MQ2008.

MQ2008								
	Methods vs. Dataset	MAP	NDCG@10	RR@10	ERR@10	P@10	Winning Rate	
	ES-Rank Gaussian	0.4745	0.5020	0.5287	0.9900	0.2793	1	
	Multi-ES-Rank 1 Gaussian Multi-ES-Rank 2 Gaussian	0.4798 0.4781	0.5022 0.5041	0.5460 0.5492	0.0984 0.0991	0.2763 0.2757	0	
	Multi-ES-Rank 3 Gaussian	0.486	0.5088	0.5551	0.0998	0.2763	4	
	ES-Rank Cauchy	0.4649	0.4832	0.5121	0.0915	0.2685	1	
	Multi-ES-Rank 1 Cauchy	0.4403	0.4713	0.5074	0.0913	0.2683	0	
	Multi-ES-Rank 2 Cauchy	0.4512	0.4822	0.5179	0.0939	0.2723	0	
Training	Multi-ES-Rank 3 Cauchy	0.456	0.4856	0.5262	0.0940	0.2725	4	
manning	ES-Rank Levy	0.4827	0.5010	0.5479	0.0996	0.2785	3	
	Multi-ES-Rank 1 Levy	0.4776	0.5032	0.5508	0.0995	0.2753	1	
	Multi-ES-Rank 2 Levy	0.4789	0.5037	0.5440	0.0987	0.2761	1	
	Multi-ES-Rank 3 Levy	0.4728	0.4972	0.5385	0.0962	0.2734	0	
	ES-Rank uniform	0.4509	0.4756	0.5333	0.0943	0.2712	2	
	Multi-ES-Rank 1 uniform	0.4419	0.4749	0.5013	0.0910	0.2693	0	
	Multi-ES-Rank 2 uniform	0.4518	0.4830	0.5184	0.0932	0.2704	3	
	Multi-ES-Rank 3 uniform	0.4461	0.4765	0.5155	0.0926	0.2676	0	

MQ2008								
	Methods vs. Dataset	MAP	NDCG@10	RR@10	ERR@10	P@10	Winning Rate	
	ES-Rank Gaussian Multi-ES-Rank 1 Gaussian Multi-ES-Rank 2 Gaussian Multi-ES-Rank 3 Gaussian ES-Rank Cauchy	0.5462 0.5344 0.5392 0.5255 0.5118	0.5613 0.5663 0.5728 0.5572 0.5307	0.5750 0.6093 0.6113 0.6067 0.5528	0.1013 0.1047 0.1056 0.1016 0.0959	0.2787 0.2768 0.2799 0.2749 0.2698	1 0 4 0	
Validation	Multi-ES-Rank 1 Cauchy Multi-ES-Rank 2 Cauchy Multi-ES-Rank 2 Cauchy ES-Rank Levy Multi-ES-Rank 1 Levy Multi-ES-Rank 2 Levy Multi-ES-Rank 2 Levy ES-Rank uniform Multi-ES-Rank 1 uniform Multi-ES-Rank 2 uniform Multi-ES-Rank 2 uniform	$\begin{array}{c} 0.4863\\ 0.4982\\ 0.4992\\ 0.5353\\ 0.5262\\ 0.5226\\ 0.5148\\ 0.4939\\ 0.4920\\ 0.4889\\ 0.4888\end{array}$	0.5256 0.5333 0.5319 0.5636 0.5594 0.5558 0.5501 0.5373 0.5288 0.5252 0.5208	$\begin{array}{c} 0.5486\\ 0.5574\\ 0.5574\\ 0.5517\\ 0.6010\\ 0.5885\\ 0.5937\\ 0.5765\\ 0.5765\\ 0.5589\\ 0.5455\\ 0.5368\\ 0.5460\end{array}$	0.0941 0.0975 0.0963 0.1059 0.1013 0.1015 0.0991 0.0980 0.0958 0.0952	0.2698 0.2710 0.2717 0.2812 0.2787 0.2761 0.2787 0.2723 0.2710 0.2704	0 3 1 5 0 0 0 5 0 0	
Testing	ES-Rank Gaussian Multi-ES-Rank 1 Gaussian Multi-ES-Rank 2 Gaussian Multi-ES-Rank 2 Gaussian ES-Rank Cauchy Multi-ES-Rank 1 Cauchy Multi-ES-Rank 2 Cauchy Multi-ES-Rank 2 Cauchy ES-Rank Levy Multi-ES-Rank 3 Cauchy ES-Rank Levy Multi-ES-Rank 1 Levy Multi-ES-Rank 2 Levy Multi-ES-Rank 3 Levy ES-Rank uniform Multi-ES-Rank 1 uniform Multi-ES-Rank 2 uniform Multi-ES-Rank 2 uniform	$\begin{array}{c} 0.4338\\ \hline 0.455\\ 0.4626\\ 0.4521\\ 0.4599\\ 0.4526\\ 0.4372\\ 0.4416\\ 0.4452\\ 0.4505\\ 0.4509\\ 0.4492\\ 0.4521\\ 0.4455\\ 0.4440\\ 0.4410\\ 0.4343 \end{array}$	$\begin{array}{c} 0.3308 \\ \hline 0.4849 \\ 0.4848 \\ 0.4807 \\ 0.4862 \\ 0.4643 \\ 0.4630 \\ 0.4655 \\ 0.4728 \\ 0.4728 \\ 0.4731 \\ 0.4732 \\ 0.4793 \\ 0.4731 \\ 0.4617 \\ 0.4675 \\ 0.4701 \\ 0.4667 \end{array}$	$\begin{array}{c} 0.54399\\ \hline 0.5056\\ 0.5460\\ 0.5492\\ 0.5551\\ 0.4690\\ 0.5074\\ 0.5179\\ 0.5262\\ 0.4945\\ 0.5268\\ 0.5440\\ 0.5385\\ 0.5440\\ 0.5385\\ 0.4888\\ 0.5013\\ 0.5184\\ 0.5155\end{array}$	0.0939 0.0939 0.0985 0.0991 0.0998 0.9130 0.0913 0.0939 0.0940 0.0956 0.0995 0.0987 0.0962 0.0946 0.0910 0.0932 0.0926	$\begin{array}{c} 0.2710\\ \hline 0.2630\\ 0.2636\\ 0.2630\\ 0.2698\\ 0.2611\\ 0.2617\\ 0.2655\\ 0.2636\\ 0.2643\\ 0.2623\\ 0.2623\\ 0.2655\\ 0.2636\\ 0.2623\\ 0.2617\\ 0.2636\\ 0.2611\\ \end{array}$	0 0 1 0 4 1 0 1 3 1 2 1 1 2 0 3 0	

Table A3. Cont.

 Table A4. The performance results of the proposed methods using MQ2008.

MQ2008								
Me	ethod Used/Winning Eva	luation Rate		Winning Rate				
		Caussian	Evolving	2				
		Gaussian	Predictive	0				
		Cauchy	Evolving	2				
Single-objective FS-Rank		Caucity	Predictive	1				
Shigle Objective ES Rank		Levy	Evolving	8				
		Levy	Predictive	1				
		I I: (	Evolving	7				
		Uniform	Predictive	2				
		Countier	Evolving	0				
		Gaussian	Predictive	1				
		Cauchy	Evolving	0				
	Matha J 1		Predictive	0				
	Method 1	Lovy	Evolving	1				
		Levy	Predictive	2				
		Uniform	Evolving	0				
Multiphiosting EC Bank			Predictive	0				
Multiobjective ES-Kalik		Coursian	Evolving	4				
		Gaussian	Predictive	0				
		Cauchy	Evolving	3				
	Matha 4.2	Caucity	Predictive	1				
	Method 2	Lovy	Evolving	1				
		Levy	Predictive	1				
		TT : C	Evolving	3				
		Uniform	Predictive	3				

## Table A4. Cont.

MQ2008							
Method Used/Winning Evaluation Rate							
		Coursian	Evolving	4			
	Method 3	Gaussian	Predictive	4			
		Cauchy	Evolving	5			
Multiphipative FC Dank		Caucity	Predictive	3			
Multiobjective ES-Kank		Levy	Evolving	0			
			Predictive	1			
		<b>TT</b> 10	Evolving	0			
		Uniform	Predictive	0			

# Table A5. Average results of the proposed methods using MQ2007.

MQ2007							
	Methods vs. Dataset	MAP	NDCG@10	RR@10	ERR@10	P@10	Winning Rate
	ES-Rank Gaussian	0.4436	0.4234	0.5664	0.0983	0.3777	2
	Multi-ES-Rank 1 Gaussian	0.4487	0.4271	0.5656	0.0994	0.3641	3
	Multi-ES-Rank 2 Gaussian	0.4435	0.4234	0.5646	0.0989	0.3661	0
	Multi-ES-Rank 3 Gaussian	0.4444	0.4219	0.5540	0.0980	0.3649	0
	ES-Rank Cauchy	0.4327	0.4111	0.5419	0.0960	0.3469	1
	Multi-ES-Rank 1 Cauchy	0.4322	0.4117	0.5421	0.0964	0.3548	0
	Multi-ES-Rank 2 Cauchy	0.4317	0.4127	0.5429	0.0969	0.3576	4
Training	Multi-ES-Rank 3 Cauchy	0.4252	0.4041	0.5385	0.0945	0.3524	0
manning	ES-Rank Levy	0.4522	0.4292	0.5488	0.1000	0.3728	1
	Multi-ES-Rank 1 Levy	0.4523	0.4325	0.5654	0.0997	0.3688	1
	Multi-ES-Rank 2 Levy	0.4531	0.4295	0.5662	0.0981	0.3706	1
	Multi-ES-Rank 3 Levy	0.4526	0.4322	0.5696	0.0986	0.3755	2
	ES-Rank uniform	0.4350	0.4195	0.5331	0.0936	0.3414	1
	Multi-ES-Rank 1 uniform	0.4374	0.4175	0.5473	0.0977	0.3601	3
	Multi-ES-Rank 2 uniform	0.4196	0.4000	0.5344	0.0941	0.3465	0
	Multi-ES-Rank 3 uniform	0.4294	0.4104	0.5395	0.0959	0.3560	1
	ES-Rank Gaussian	0.4742	0.4520	0.5930	0.1066	0.3832	4
	Multi-ES-Rank 1 Gaussian	0.4682	0.4511	0.5818	0.1054	0.3714	0
	Multi-ES-Rank 2 Gaussian	0.4744	0.4515	0.5876	0.1047	0.3755	1
	Multi-ES-Rank 3 Gaussian	0.472	0.4499	0.5925	0.1049	0.3714	0
	ES-Rank Cauchy	0.4546	0.4336	0.5535	0.1006	0.3575	0
	Multi-ES-Rank 1 Cauchy	0.4563	0.4325	0.5588	0.1005	0.3563	0
	Multi-ES-Rank 2 Cauchy	0.4573	0.4383	0.5715	0.1027	0.3593	5
\$7.1.1	Multi-ES-Rank 3 Cauchy	0.4536	0.4293	0.5675	0.1008	0.3525	0
Validation	ES-Rank Levy	0.4742	0.4628	0.5931	0.1081	0.3838	3
	Multi-ES-Rank 1 Levy	0.476	0.4541	0.6115	0.1071	0.3791	2
	Multi-ES-Rank 2 Levy	0.4726	0.4524	0.5848	0.1046	0.3802	0
	Multi-ES-Rank 3 Levy	0.4756	0.4560	0.5923	0.1072	0.3805	0
	ES-Rank uniform	0.4601	0.4439	0.5495	0.0977	0.3525	0
	Multi-ES-Rank 1 uniform	0.4664	0.4474	0.5853	0.1053	0.3643	4
	Multi-ES-Rank 2 uniform	0.4437	0.4218	0.5458	0.0983	0.3504	0
	Multi-ES-Rank 3 uniform	0.4574	0.4357	0.5616	0.1013	0.3581	1
	ES-Rank Gaussian	0.4774	0.4650	0.5683	0.1094	0.4006	4
	Multi-ES-Rank 1 Gaussian	0.4816	0.4623	0.5656	0.0994	0.3887	0
	Multi-ES-Rank 2 Gaussian	0.4688	0.4528	0.5646	0.0989	0.3896	0
	Multi-ES-Rank 3 Gaussian	0.4818	0.4623	0.5540	0.0980	0.3929	1
	ES-Rank Cauchy	0.4561	0.4378	0.5524	0.1028	0.3694	2
	Multi-ES-Rank 1 Cauchy	0.4621	0.4375	0.5421	0.0964	0.3723	1
	Multi-ES-Rank 2 Cauchy	0.4611	0.4395	0.5429	0.0964	0.3735	2
т. ( <sup>.</sup>	Multi-ES-Rank 3 Cauchy	0.4534	0.4326	0.5385	0.0945	0.3726	0
lesting	ES-Rank Levy	0.4833	0.4590	0.5640	0.1074	0.3900	1
	Multi-ES-Rank 1 Levy	0.4825	0.4669	0.5654	0.0997	0.3970	0
	Multi-ES-Rank 2 Levy	0.4833	0.4675	0.5662	0.0981	0.3976	1
	Multi-ES-Rank 3 Levv	0.4844	0.4678	0.5696	0.0986	0.3958	3
	ES-Rank uniform	0.4635	0.4448	0.5462	0.1001	0.3586	1
	Multi-ES-Rank 1 uniform	0.4706	0.4472	0.5473	0.0977	0.3777	4
	Multi-ES-Rank 2 uniform	0.4509	0.4301	0.5344	0.0941	0.3676	0
	Multi-ES-Rank 3 uniform	0.4622	0.4385	0.5395	0.0959	0.3732	ů 0

## References

- 1. Li, H. Theory of learning to rank. In *Learning to Rank for Information Retrieval and Natural Language Processing*, 2nd ed.; Springer International Publishing: Cham, Switzerland, 2015; pp. 81–86.
- 2. Manning, C.D.; Raghavan, P.; Schütze, H. Introduction to Information Retrieval; Cambridge University Press: Cambridge, UK, 2008.
- 3. Urbano, J. Test collection reliability: A study of bias and robustness to statistical assumptions via stochastic simulation. *Inf. Retr. J.* **2016**, *19*, 313–350. [CrossRef]
- Momma, M.; Dong, C.; Chen, Y. Multi-objective Ranking with Directions of Preferences. In Proceedings of the 45th International ACM SIGIR Conference in Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022.
- Drucker, H.; Shahrary, B.; Gibbon, D.C. Support vector machines: Relevance feedback and information retrieval. *Inf. Process. Manag.* 2002, 38, 305–323. [CrossRef]
- Al-Tashi, Q.; Abdulkadir, S.J.E. Approaches to multi-objective feature selection: A systematic literature review. *IEEE Access* 2020, 8, 125076–125096. [CrossRef]
- Xue, B.; Zhang, M.; Browne, W.N.; Yao, X. A survey on evolutionary computation approaches to feature selection. *IEEE Trans. Evol. Comput.* 2016, 20, 606–626. [CrossRef]
- 8. Svore, K.M.; Volkovs, M.N.; Burges, C.J.C. Learning to Rank with Multiple Objective Functions. In Proceedings of the 20th International World Wide Web Conference, New York, NY, USA, 28 March–1 April 2011.
- 9. Li, W.; Chai, Z.; Tang, Z. A decomposition-based multi-objective immune algorithm for feature selection in learning to rank. *Knowl.-Based Syst.* **2021**, 234, 107577. [CrossRef]
- Kundu, P.P.; Mitra, S. Multi-objective optimization of shared nearest neighbor similarity for feature selection. *Appl. Soft Comput.* 2015, 37, 751–762. [CrossRef]
- 11. Yong, Z.; Gong, D.W.; Sun, X.Y.; Guo, Y.N. A PSO-based multi-objective multi-label feature selection method in classification. *Sci. Rep.* **2017**, *7*, 376.
- 12. Das, A.; Das, S. Feature weighting and selection with a pareto optimal tradeoff between relevancy and redundancy. *Pattern Recognit. Lett.* **2017**, *88*, 12–19. [CrossRef]
- 13. Cheng, F.; Guo, W.; Zhang, X. Mofsrank: A multi-objective evolutionary algorithm for feature selection in learning to rank. *Complexity* **2018**, 2018, 14. [CrossRef]
- Mahapatra, D.; Dong, C.; Chen, Y.; Meng, D.; Momma, M. Multi-Label Learning to Rank through Multi-Objective Optimization. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Long Beach, CA, USA, 6–10 September 2023.
- Pang, L.; Xu, J.; Ai, Q.; Lan, Y.; Cheng, X.; Wen, J. Setrank: Learning a Permutation-invariant Ranking Model for Information Retrieval. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 25–30 July 2020.
- 16. Ibrahim, O.A.S.; Silva, D.L. An evolutionary strategy with machine learning for learning to rank in information retrieval. *Soft Comput.* **2018**, *22*, 3171–3185. [CrossRef]
- 17. Yan, L.; Qin, Z.; Wang, X.; Bendersky, M.; Najork, M. Scale Calibration of Deep Ranking Models. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 14–18 August 2022.
- Singh, A.; Joachims, T. Policy Learning for Fairness in Ranking. In Proceedings of the 2019 Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
- Morik, M.; Singh, A.; Hong, J.; Joachims, T. Controlling Fairness and Bias in Dynamic Learning-to-Rank. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 25–30 July 2020.
- Ibrahim, O.A.S.; Younis, E.M.G. Hybrid online offline learning to rank using simulated annealing strategy based on dependent click model. *Knowl. Inf. Syst.* 2022, 64, 2833–2847. [CrossRef]
- Loshchilov, L. A Computationally Efficient Limited Memory Cmaes for Large Scale Optimization. In Proceedings of the Genetic and Evolutionary Computation Conference, Vancouver, BC, Canada, 12–16 July 2014.
- 22. Qin, T.; Liu, T. Introducing LETOR 4.0 datasets. arXiv 2013, arXiv:1306.2597v1.
- 23. Liu, T. The LETOR dataset. In Learning to Rank for Information Retrieval; Springer: Berlin, Germany, 2011; pp. 133–143.
- 24. Qin, T.; Liu, T.; Xu, J.; Li, H. Letor: A benchmark collection for research on learning to rank for information retrieval. *Inf. Retr.* **2010**, *13*, 346–374. [CrossRef]
- Ibrahim, O.A.S.; Landa-Silva, D. Es-rank: Evolution Strategy Learning to Rank Approach. In Proceedings of the 32nd ACM SIGAPP Symposium on Applied Computing, Marrakech, Morocco, 4–6 April 2017.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.