Combining Variable Neighbourhood with Simulated Annealing for Learning to Rank Problem

Osman Ali Sadek Ibrahim (osmaneg200@gmail.com)
Minia University Faculty of Science  https://orcid.org/0000-0001-9254-3093

Research Article

Keywords: Learning to Rank, Variable Neighbourhood Search, Random Generator Numbers, Information Retrieval, MSLR-WEB10K, Yahoo

Posted Date: November 25th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2293850/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License
Combining Variable Neighbourhood with Simulated Annealing for Learning to Rank Problem

Osman Ali Sadek Ibrahim

Received: date / Accepted: date

Abstract Variable Neighbourhood Search (VNS) is a problem-solving technique that improves heuristic solutions. The solutions are built around incremental adjustments to neighboring solutions. Changes are done during the climbing phase to get local optimal solutions, followed by a stochastic phase to achieve global optimum solutions. A range of mutation step-sizes are used in the exploration and exploitation methods. The function’s purpose is to choose the best offspring to pass on to the next developing generation. As a consequence, this paper changes the Offspring solution in the following iteration applying a version of VNS based on four random probability distributions. Once the cooling temperature is met, it is capable of accepting a bad solution. Variable Neighborhood Annealing is a novel approach in Learning to Rank (LTR) (VNA). Each Offspring ranking model solution is built from a single probability distribution during the mutation phase (all mutation step-sizes made by only one probability distribution for each Neighbourhood candidate). Based on the results, we may infer that the VNA approach outperformed contemporary research on Evolutionary and Machine Learning methodologies. In the studies, we used datasets from Yahoo, Microsoft Bing Search (MSLR-WEB10K), and LETOR 4 (MQ2008, MQ2007).

Keywords Learning to Rank · Variable Neighbourhood Search · Random Generator Numbers · Information Retrieval · MSLR-WEB10K · Yahoo

1 Introduction

The variable Neighbourhood Search (VNS) algorithm is classified as a Local Search (LS) algorithm. It does, however, avoid the issue of LS being trapped in local optimal solutions. It is employed in order to locate the greatest global solutions [Mladenović and Hansen, 1997]. More exploration for examining various neighbourhood solutions with non-deterministic and regular step-sizes in evolving iteration can result in this. Local search techniques for optimization problems have been used to iterate locally on an initial solution, enhancing the value of the objective function each time, resulting in local optimal solutions. However, after obtaining local optimum solutions, an improved solution x’ in the neighbourhood N(x) of the current solution x can be obtained at each iteration. As a result, several general heuristics (or metaheuristics) have been proposed in recent years to diversify this strategy and prevent it from becoming stuck in local optima with

Dr. Osman Ali Sadek Ibrahim
A Corresponding Author,
An Assistant Professor,
Computer Science Department, Minia University, Egypt.
E-mail: osman.ibrahim@mu.edu.eg
bad solutions. Examples of these techniques which were examined in the previous studies for other problem domains are: genetic algorithms (GAs) [Michalewicz, 1996], evolutionary strategies (ES) [Kramer, 2016], evolutionary programming (EP) [Fogel, 1999], and evolutionary algorithms (EAs) [Nannen et al., 2008].

Previous studies did considerable study on optimum and near-optimal solutions using heuristics and metaheuristics approaches. Despite this, only a few of these optimization techniques for the Learning to Rank (LTR) issue in Information Retrieval (IR) have been presented [Ibrahim, 2022, Ibrahim and Younis, 2022, Ibrahim and Landa-Silva, 2018]. This paper describes a novel VNS approach for improving the performance of ranking systems. For the top ten documents retrieved, we used two metrics: Mean Average Precision (MAP) and Normalized Cumulative Gain (NDCG@10). The suggested approach for modifying the present solution to discover the best Neighbourhood has never been applied previously in the optimization issue arena. In addition, the VNS algorithms have never been applied to the LTR issue.

The main Contributions of this paper are as follows:

– In continuous optimization research, we propose a novel methodology for the sequence Variable Neighbourhood Annealing approach. This is accomplished by employing five probability distributions as random number generators for the Neighbourhood search to achieve the optimal solutions. It also combines Simulated Annealing and Variable Neighborhood Search to accept bad results with a move procedure once the cooling temperature has been reached.
– Applying Variable Neighbourhood Search approach for the first time in the Learning to Rank problem by the proposed method.
– Using a comparative study of the proposed method and recent research studies in the domain of ranking. This is accomplished by utilising the largest and most recent Learning to Rank datasets available.
– Providing the Java Archive Packages for the proposed method for research reproducibility.

This paper is organized as follows. Section 2 discusses the related research existing in previous studies. Section 3 presents the proposed VNS approaches, while section 4 presents the results obtained using Yahoo, MSLR-WEB10K and LETOR 4 (MQ2008 and MQ2007) datasets. Section 5 provides the conclusions and recommendations for future research.

2 Related Work

There are three main types of LTR methodologies [Liu, 2009, Cao et al., 2007, Li, 2014] which are the pointwise approaches, pairwise methods, and listwise techniques. These categories are based on evaluation steps of the metric function or loss function. Each unique item (query-document pair) is considered as a learning sample in the pointwise techniques. This category is illustrated by various types of algorithms. Linear Regression (LR) [Yan and Su, 2009], Boosting [Freund et al., 2003], Gradient Boosted Regression Trees (GBRT or MART) [Friedman, 2001, Mohan et al., 2011], and Random Forest (RF) [Breiman, 2001] are examples of the pointwise methods. The Pairwise methods use pair or two query-document pairs for the same query as the learning instance and incorporates a loss or fitness function. RankNET (Rank Neural Net) [Burges et al., 2005], RankBoost, and SVMRank [Li, 2014] are three examples of Pairwise methods. The learning instance in the listwise technique is the list of query-document instances for each query as the entire retrieved list of contents. ListNET (Listwise Neural Net) [Cao et al., 2007], Coordinate Ascent [Metzler and Bruce Croft, 2007], AdaRank [Xu and Li, 2007], and RankGPES [Islam, 2013] are some examples of listwise methods. The aforementioned approaches are proposed after previous research for the modelling of user clicks.
The new trends of research in LTR is called Online LTR. This research uses implicit relevance labels to imitate user clicks and evaluate how well the Online LTR algorithm perform. This is because the actual user clicks may suffer from several bias issues which were mentioned in these studies [Hofmann et al., 2013, Schuth et al., 2013].

On the contrary, a research study has been conducted to prove that learning from explicit relevance label outperforms the learning from implicit relevance feedback in online LTR methods using Click simulation models [Ibrahim, 2017, Ibrahim and Younis, 2022]. From another perspective, research studies have been done for user click bias corrections to achieve better performance on ranking search engines webpages with improved relationship between dataset features with its relevance feedback labels [Oosterhuis, 2020, Vardasbi et al., 2021]. In [Vardasbi et al., 2021], two well-known LTR algorithms from the literature have been used for various aspects of bias corrections. These algorithms are Deep Neural Network and LambdaMART which are outperformed previous studies.

In this research, we present a methodology for combining sequence Variable Neighbourhood with Simulated Annealing approach. This is done using five probability distributions as random numbers generators for mutation step-size of Neighbourhood search for optimal solutions. Moreover, applying Variable Neighbourhood Search approach for the first time in the Learning to Rank problem by the proposed methods. Furthermore, comparing among the method proposed and recent research in the ranking problem domain [Oosterhuis, 2020, Vardasbi et al., 2021, Ibrahim, 2017, Ibrahim and Younis, 2022, Ibrahim, 2022]. This was done using two of the largest and most recent available Learning to Rank datasets and two well-known datasets from TREC research tracks.

3 The Proposed Approaches

A novel Variable Neighborhood Annealing (VNA) technique is used in this paper. The VNA method operates in the following manner. A list of various neighbourhood Offspring solutions (OffNeighborhood) are first generated, and each one is then evaluated one by one in the order specified. Let Neighbourhood = OffNeighbourhood1, OffNeighbourhood2, ..., OffNeighbourhoodn be a i-th generation of Offspring Neighbourhood generation of Offspring neighbourhood solutions, with the neighbourhood ranking models specified as the neighbourhood ranking models to the current Parent ranking solution. Fitness metric evaluation refers to the process of comparing their accuracy to the best current solution accuracy for this evolving iteration. Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain at top-10 documents retrieved (NDCG@10) were two well-known fitness evaluation accuracy used in this process.

Starting from a given solution \( X \) and while \( \text{MaxGenerations} \) is the maximum number of evolving iterations, the VNA steps iterative seek to explore the neighbourhood ranking model solution specified in each evolving iteration based on evolving iterations \( 1 \leq I \leq \text{MaxGenerations} \) one by one in the set order. The VNA process resumes search in the current neighbourhood structure (by replacing the current solution with the best-performed neighbourhood solution in the previous evolving iteration) of the new offspring solution as soon as an improvement of the current parent solution in a particular neighbourhood structure occurs.

Algorithm 1 provides the VNA’s sequential steps for employing the most effective search methodology. The VNA algorithm generates the neighbourhood solution by mutating the current solution using one of five probability random number generators: Gaussian, Cauchy, Levy, Uniform, and traditional distributions. For each neighbourhood solution, several random mutation step-sizes from a single probability distribution are used. These probability distributions have
Algorithm 1: VNA-Rank: Sequential Variable Neighbourhood Ascent Ranking Technique

Input: A training set \( \phi(q,d) \) 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 \((Gen_i \in X)\) do
3 \( Gen_i = 0.5 \);
4 end
5 for \(i = 1\) to \(\text{MaxGenerations}\) do
6 Choose number of genes to mutate \( R \) at random from 1 to \( M \) in \( X \);
7 To generate Neighbourhood Solution \( Neighbourhood_i \) according to the method chosen.;
8 if \( (\text{Fitness}(X,\phi(q,d)) < \text{Fitness}(Neighbourhood_i,\phi(q,d))) \) then
9 \( X = Neighbourhood_i \);
10 else
11 \( Neighbourhood_i = X; \)
12 end
13 if \( ((\text{Fitness}(Neighbourhood_i,\phi(q,d)) < \text{Fitness}(X,\phi(q,d))) \) \&\&\& Cooling temp is reach \) then
14 \( X = Neighbourhood_i \);
15 else
16 end
17 end
18 return the linear ranking function \( F(q,d) = X^T \cdot \phi(q,d) = W^T \cdot \phi(q,d) \), that is \( X \) at the end of the \( \text{MaxGenerations} \) contains the evolved vector \( W \) of \( M \) feature weights, \( T \) indicates the transpose

been used in a number of research studies published in the literature to solve other optimization problems. [Ibrahim, 2022, Zheng et al., 2021, Kramer, 2016, Michalewicz, 1996]. The java archive packages for the novel Variable Neighbourhood Research method is provided in VNA.

4 Results

The results of an experimental study comparing the performance of the proposed VNS approach to evolutionary and machine learning techniques are presented in this section. Benchmark datasets containing training, validation, and test sets were used to build the ranking model and assess the performance of the LTR methods. To find the best ranking model, the LTR technique is used first for model training. The predictive performance of the LTR algorithm is then measured by evaluating the ranking model’s efficiency using the test set.

4.1 Benchmark Datasets and Evaluation Fitness Metric

The studies in this research utilized the benchmark datasets: MSLR-WEB10K and LETOR 4 (MQ2007 and MQ2008) [Qin and Liu, 2013, Chapelle and Chang, 2011, Liu, 2011, Qin et al., 2010]. The characteristics of these datasets are described in Table 1. The larger LTR datasets available for research are the Yahoo and Microsoft Bing Search datasets (MSLR-WEB10K) and they conducted in this research. Low-level features such term frequency and inverse document frequency
Table 1 Properties of the benchmark data used in the experimental study.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Queries</th>
<th>Query-Document Pairs</th>
<th>Features</th>
<th>Relevance Labels</th>
<th>No. of Folds</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQ2007</td>
<td>1692</td>
<td>69623</td>
<td>46</td>
<td>{0, 1, 2}</td>
<td>5</td>
</tr>
<tr>
<td>MQ2008</td>
<td>784</td>
<td>15211</td>
<td>46</td>
<td>{0, 1, 2}</td>
<td>5</td>
</tr>
<tr>
<td>Yahoo Set1</td>
<td>29921</td>
<td>709877</td>
<td>519</td>
<td>{0, 1, 2, 3, 4}</td>
<td>1</td>
</tr>
<tr>
<td>MSLR-WEB10K</td>
<td>10000</td>
<td>1200192</td>
<td>136</td>
<td>{0, 1, 2, 3, 4}</td>
<td>5</td>
</tr>
</tbody>
</table>

of the document terms included in each query-document pair. For each document part, the low-level features were evaluated (title, anchor, body and whole) for information content. High-level features that reflect how closely the query and the documents match each other also exist in each learning instance. The language model with absolute discounted smoothing (LMIR.ABS), language model with jelinek-mercer smoothing (LMIR.JM), language model with bayesian smoothing using dirichlet priors (LMIR.DIR), and user click features which illustrated in the recent research in IR models are also features existing in each dataset instance [Qin and Liu, 2013, Chapelle and Chang, 2011, Liu, 2011, Qin et al., 2010]. Each query contains a number of relevant and irrelevant documents attached to it, called query-document pairs. The relevance label specifies the levels of relevance for the documents to the query (query-document relationship). The relevance labels typically have values of 0 for irrelevant, 1 for partially relevant, and 2 for totally relevant. The MSLR-WEB10K dataset, created by the Bing search engine, and Yahoo search engine dataset have relevance values ranging from 0 (irrelevant) to 4 (perfectly relevant), while LETOR 4 datasets have only 3 degree of relevance ranging from irrelevance to totally relevance.

This paper uses two evaluation fitness metrics to support the research results obtained which are Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain at top-10 documents retrieved (NDCG@10). Both evaluation metrics have been used extensively in the previous studies such as in [Oosterhuis, 2020, Vardasbi et al., 2021, Chapelle and Chang, 2011, Ibrahim and Landas-Silva, 2018]. The MAP metric just evaluates whether the query-document retrieved is relevant or not, not the graded relevance levels of each document that is retrieved. Instead of focusing on the top-k query-document pairs, MAP assesses the average precision over the entire set of search results. The graded relevance level of each pair of retrieved top-k query documents is factored by the NDCG@K metric.

In the following subsection, we present the analysis of the results obtained. The analysis is divided into both researchers’ point-of-views. Firstly, from Metaheuristic and Evolutionary Computation algorithms analysis point-of-view and Machine Learning point-of-view. Thus, this paper includes the VNA abilities in evolving optimal ranking models when applied to the Training and Validating dataset besides their capabilities for predictive results on the test dataset. The method used in comparison to the previous studies are the most recent approach in the literature which outperformed other available methods [Ibrahim, 2017, Ibrahim and Younis, 2022]. (1+1)-Evolutionary Strategy Learning to Rank (ES-Rank) outperformed fourteen evolutionary and machine learning methods as demonstrated in [Ibrahim, 2017]. While extension for it by combination with Simulated Annealing (SAS-Rank) or Gradient Mutation Step-size added (EGS-Rank) have been proposed in recent research for better performance [Ibrahim, 2022, Ibrahim and Younis, 2022]. In addition, other researchers provided bias corrections in relevance feedback for better performance on the well-known LambdaMART and Deep Neural Network (DNN) among methods provided in [Oosterhuis, 2020, Vardasbi et al., 2021]. These approaches with their variations were included in the comparative studies with the proposed methods.
4.2 Result analysis and Discussion

Table 2  Algorithms Average Training Performance Applied on 4 Datasets Using MAP Fitness Evaluation Metric

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Yahoo Set1</th>
<th>MSLR-WEB10K</th>
<th>MQ2008</th>
<th>MQ2007</th>
<th>Winning Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES-Rank Gaussian</td>
<td>0.84</td>
<td>0.5639</td>
<td>0.4721</td>
<td>0.4494</td>
<td>0</td>
</tr>
<tr>
<td>ES-Rank Cauchy</td>
<td>0.8327</td>
<td>0.4953</td>
<td>0.4446</td>
<td>0.4319</td>
<td>0</td>
</tr>
<tr>
<td>ES-Rank Levy</td>
<td>0.8404</td>
<td>0.5827</td>
<td>0.4771</td>
<td>0.4526</td>
<td>0</td>
</tr>
<tr>
<td>ES-Rank Uniform</td>
<td>0.8304</td>
<td>0.4818</td>
<td>0.4570</td>
<td>0.4213</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Gaussian</td>
<td>0.8334</td>
<td>0.5811</td>
<td>0.4810</td>
<td>0.4558</td>
<td>1</td>
</tr>
<tr>
<td>SAS-Rank Cauchy</td>
<td>0.8336</td>
<td>0.5751</td>
<td>0.4789</td>
<td>0.4499</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Levy</td>
<td>0.8278</td>
<td>0.5684</td>
<td>0.449</td>
<td>0.4515</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Uniform</td>
<td>0.8376</td>
<td>0.5871</td>
<td>0.4760</td>
<td>0.4526</td>
<td>1</td>
</tr>
<tr>
<td>EGS-Rank Gaussian</td>
<td>0.8365</td>
<td>0.5380</td>
<td>0.4567</td>
<td>0.4484</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Cauchy</td>
<td>0.8303</td>
<td>0.5167</td>
<td>0.4698</td>
<td>0.4490</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Levy</td>
<td>0.8271</td>
<td>0.5367</td>
<td>0.4555</td>
<td>0.4353</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Uniform</td>
<td>0.8353</td>
<td>0.5125</td>
<td>0.4708</td>
<td>0.4334</td>
<td>0</td>
</tr>
<tr>
<td>VNA</td>
<td><strong>0.8598</strong></td>
<td><strong>0.5840</strong></td>
<td><strong>0.5022</strong></td>
<td>0.4541</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 3  Algorithms Average Validation Performance Applied on 4 Datasets Using MAP Fitness Evaluation Metric

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Yahoo Set1</th>
<th>MSLR-WEB10K</th>
<th>MQ2008</th>
<th>MQ2007</th>
<th>Winning Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES-Rank Gaussian</td>
<td>0.8349</td>
<td>0.5763</td>
<td>0.5301</td>
<td>0.4799</td>
<td>0</td>
</tr>
<tr>
<td>ES-Rank Cauchy</td>
<td>0.8269</td>
<td>0.5041</td>
<td>0.4918</td>
<td>0.457</td>
<td>0</td>
</tr>
<tr>
<td>ES-Rank Levy</td>
<td>0.8353</td>
<td>0.5913</td>
<td>0.5474</td>
<td>0.4818</td>
<td>1</td>
</tr>
<tr>
<td>ES-Rank Uniform</td>
<td>0.8241</td>
<td>0.4910</td>
<td>0.5081</td>
<td>0.4440</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Gaussian</td>
<td>0.8266</td>
<td>0.5889</td>
<td>0.5196</td>
<td>0.4726</td>
<td>1</td>
</tr>
<tr>
<td>SAS-Rank Cauchy</td>
<td>0.8268</td>
<td>0.5847</td>
<td>0.5185</td>
<td>0.4711</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Levy</td>
<td>0.834</td>
<td>0.5758</td>
<td>0.4841</td>
<td>0.4698</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Uniform</td>
<td>0.8311</td>
<td>0.5984</td>
<td>0.5031</td>
<td>0.4603</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Gaussian</td>
<td>0.8317</td>
<td>0.5458</td>
<td>0.5219</td>
<td>0.4793</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Cauchy</td>
<td>0.8245</td>
<td>0.5263</td>
<td>0.5267</td>
<td>0.4654</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Levy</td>
<td>0.8202</td>
<td>0.5441</td>
<td>0.5076</td>
<td>0.4588</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Uniform</td>
<td>0.8283</td>
<td>0.5217</td>
<td>0.5078</td>
<td>0.4599</td>
<td>0</td>
</tr>
<tr>
<td>VNA</td>
<td><strong>0.8457</strong></td>
<td><strong>0.6029</strong></td>
<td><strong>0.5437</strong></td>
<td><strong>0.5010</strong></td>
<td>3</td>
</tr>
</tbody>
</table>

This subsection includes an analysis of the results obtained for comparing the proposed method VNA with other available evolutionary computation methods. The results of the evolving procedure are applied on training and validation data and the predictive results are applied to testing data. Since the comparative studies between the evolutionary computation or metaheuristic methods have been done in other optimization problem domains in the stage of evolving optimal or near-optimal solutions on the training dataset rather than predictive results obtained by the
optimal solution. Thus, the paper includes both evolving results applied on training and validating datasets besides the results obtained with predictive evaluation on test data. These results are demonstrated in tables 2 to 7. Then, this paper compares the proposed approach performance with recent and well-known methods in previous studies [Oosterhuis, 2020, Vardasbi et al., 2021]. The results for that are demonstrated in table 8.

In table 2 to 7, the bold results indicates the best evaluation fitness metric obtained on the dataset. Each of these tables has the last column which contains the number of winning best performances by each method. Generally, VNA method outperformed the other methods. The total number of best performances using VNA is 16 (8 MAP and 8 NDCG@10) values out of 24 evaluation metric values. On the other hand, the second method for obtaining the best performance is

### Table 4 Algorithms Average Predictive Performance Applied on 4 Datasets Using MAP Fitness Evaluation Metric

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Yahoo Set1</th>
<th>MSLR-WEB10K</th>
<th>MQ2008</th>
<th>MQ2007</th>
<th>Winning Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES-Rank Gaussian</td>
<td>0.8446</td>
<td>0.5685</td>
<td>0.4515</td>
<td>0.4866</td>
<td>1</td>
</tr>
<tr>
<td>ES-Rank Cauchy</td>
<td>0.8365</td>
<td>0.4971</td>
<td>0.4354</td>
<td>0.465</td>
<td>0</td>
</tr>
<tr>
<td>ES-Rank Levy</td>
<td>0.8436</td>
<td>0.5819</td>
<td>0.4607</td>
<td>0.4853</td>
<td>0</td>
</tr>
<tr>
<td>ES-Rank Uniform</td>
<td>0.8344</td>
<td>0.4856</td>
<td>0.4468</td>
<td>0.4468</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Gaussian</td>
<td>0.8375</td>
<td>0.5799</td>
<td>0.453</td>
<td>0.4851</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Cauchy</td>
<td>0.8373</td>
<td>0.5746</td>
<td>0.4633</td>
<td>0.4757</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Levy</td>
<td>0.8380</td>
<td>0.5688</td>
<td>0.4506</td>
<td>0.4847</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Uniform</td>
<td>0.8419</td>
<td>0.5877</td>
<td>0.4582</td>
<td>0.4804</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Gaussian</td>
<td>0.8398</td>
<td>0.5391</td>
<td>0.4515</td>
<td>0.4818</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Cauchy</td>
<td>0.8339</td>
<td>0.5172</td>
<td>0.4599</td>
<td>0.4788</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Levy</td>
<td>0.8306</td>
<td>0.5390</td>
<td>0.4503</td>
<td>0.4658</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Uniform</td>
<td>0.8361</td>
<td>0.5129</td>
<td>0.4616</td>
<td>0.4621</td>
<td>0</td>
</tr>
<tr>
<td>VNA</td>
<td><strong>0.8572</strong></td>
<td><strong>0.6036</strong></td>
<td><strong>0.4856</strong></td>
<td>0.4845</td>
<td>3</td>
</tr>
</tbody>
</table>

### Table 5 Algorithms Average Training Performance Applied on 4 Datasets Using NDCG@10 Fitness Evaluation Metric

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Yahoo Set1</th>
<th>MSLR-WEB10K</th>
<th>MQ2008</th>
<th>MQ2007</th>
<th>Winning Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES-Rank Gaussian</td>
<td>0.7054</td>
<td>0.3766</td>
<td>0.4986</td>
<td>0.4275</td>
<td>0</td>
</tr>
<tr>
<td>ES-Rank Cauchy</td>
<td>0.6761</td>
<td>0.2441</td>
<td>0.4810</td>
<td>0.4126</td>
<td>0</td>
</tr>
<tr>
<td>ES-Rank Levy</td>
<td>0.7063</td>
<td>0.3837</td>
<td>0.5062</td>
<td>0.4275</td>
<td>0</td>
</tr>
<tr>
<td>ES-Rank Uniform</td>
<td>0.6740</td>
<td>0.2357</td>
<td>0.4858</td>
<td>0.4030</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Gaussian</td>
<td>0.7703</td>
<td>0.4798</td>
<td><strong>0.5585</strong></td>
<td><strong>0.4893</strong></td>
<td>2</td>
</tr>
<tr>
<td>SAS-Rank Cauchy</td>
<td>0.7549</td>
<td>0.4520</td>
<td>0.5484</td>
<td>0.4879</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Levy</td>
<td>0.7649</td>
<td>0.3883</td>
<td>0.5258</td>
<td>0.4283</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Uniform</td>
<td>0.7712</td>
<td>0.4750</td>
<td>0.5521</td>
<td>0.4901</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Gaussian</td>
<td>0.7492</td>
<td>0.4168</td>
<td>0.5500</td>
<td>0.4671</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Cauchy</td>
<td>0.7448</td>
<td>0.3498</td>
<td>0.5439</td>
<td>0.4814</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Levy</td>
<td>0.7541</td>
<td>0.4016</td>
<td>0.5337</td>
<td>0.4658</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Uniform</td>
<td>0.7440</td>
<td>0.3323</td>
<td>0.5441</td>
<td>0.4794</td>
<td>0</td>
</tr>
<tr>
<td>VNA</td>
<td><strong>0.7859</strong></td>
<td><strong>0.4966</strong></td>
<td>0.5488</td>
<td>0.4876</td>
<td>2</td>
</tr>
</tbody>
</table>
Table 6 Algorithms Average Validation Performance Applied on 4 Datasets Using NDCG@10 Fitness Evaluation Metric

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Yahoo Set1</th>
<th>MSLR-WE10K</th>
<th>MQ2008</th>
<th>MQ2007</th>
<th>Winning Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES-Rank Gaussian</td>
<td>0.7632</td>
<td>0.3784</td>
<td>0.5628</td>
<td>0.4580</td>
<td>0</td>
</tr>
<tr>
<td>ES-Rank Cauchy</td>
<td>0.6721</td>
<td>0.2490</td>
<td>0.5405</td>
<td>0.4374</td>
<td>0</td>
</tr>
<tr>
<td>ES-Rank Levy</td>
<td>0.7632</td>
<td>0.3911</td>
<td>0.5082</td>
<td>0.4586</td>
<td>0</td>
</tr>
<tr>
<td>ES-Rank Uniform</td>
<td>0.6714</td>
<td>0.2389</td>
<td>0.5362</td>
<td>0.4370</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Gaussian</td>
<td>0.7631</td>
<td>0.4585</td>
<td>0.5832</td>
<td>0.5076</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Cauchy</td>
<td>0.7538</td>
<td>0.4555</td>
<td>0.6091</td>
<td>0.5057</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Levy</td>
<td>0.7625</td>
<td>0.3969</td>
<td>0.5651</td>
<td>0.4607</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Uniform</td>
<td>0.7660</td>
<td>0.4829</td>
<td>0.6013</td>
<td>0.5101</td>
<td>1</td>
</tr>
<tr>
<td>EGS-Rank Gaussian</td>
<td>0.7438</td>
<td>0.4240</td>
<td>0.6075</td>
<td>0.4848</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Cauchy</td>
<td>0.7417</td>
<td>0.3527</td>
<td>0.6104</td>
<td>0.5037</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Levy</td>
<td>0.7495</td>
<td>0.4043</td>
<td>0.6043</td>
<td>0.4897</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Uniform</td>
<td>0.7422</td>
<td>0.3357</td>
<td>0.6017</td>
<td>0.5050</td>
<td>0</td>
</tr>
<tr>
<td>VNA</td>
<td><strong>0.7689</strong></td>
<td><strong>0.4961</strong></td>
<td><strong>0.6107</strong></td>
<td>0.5186</td>
<td><strong>3</strong></td>
</tr>
</tbody>
</table>

Table 7 Algorithms Average Predictive Performance Applied on 4 Datasets Using NDCG@10 Fitness Evaluation Metric

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Yahoo Set1</th>
<th>MSLR-WE10K</th>
<th>MQ2008</th>
<th>MQ2007</th>
<th>Winning Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES-Rank Gaussian</td>
<td>0.7073</td>
<td>0.3710</td>
<td>0.4776</td>
<td>0.4622</td>
<td>0</td>
</tr>
<tr>
<td>ES-Rank Cauchy</td>
<td>0.6790</td>
<td>0.2455</td>
<td>0.4667</td>
<td>0.4404</td>
<td>0</td>
</tr>
<tr>
<td>ES-Rank Levy</td>
<td>0.7082</td>
<td>0.3789</td>
<td>0.4793</td>
<td>0.4529</td>
<td>0</td>
</tr>
<tr>
<td>ES-Rank Uniform</td>
<td>0.6765</td>
<td>0.2390</td>
<td>0.4636</td>
<td>0.4361</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Gaussian</td>
<td>0.7726</td>
<td>0.4456</td>
<td>0.5224</td>
<td>0.5256</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Cauchy</td>
<td>0.7582</td>
<td>0.4425</td>
<td>0.5221</td>
<td>0.5157</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Levy</td>
<td>0.7650</td>
<td>0.3925</td>
<td>0.5146</td>
<td>0.4686</td>
<td>0</td>
</tr>
<tr>
<td>SAS-Rank Uniform</td>
<td>0.7740</td>
<td>0.4706</td>
<td>0.5297</td>
<td><strong>0.5191</strong></td>
<td><strong>1</strong></td>
</tr>
<tr>
<td>EGS-Rank Gaussian</td>
<td>0.7526</td>
<td>0.4147</td>
<td>0.5242</td>
<td>0.5015</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Cauchy</td>
<td>0.7481</td>
<td>0.3471</td>
<td>0.5158</td>
<td>0.5138</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Levy</td>
<td>0.7575</td>
<td>0.3957</td>
<td>0.5076</td>
<td>0.4904</td>
<td>0</td>
</tr>
<tr>
<td>EGS-Rank Uniform</td>
<td>0.7477</td>
<td>0.3300</td>
<td>0.5224</td>
<td>0.5079</td>
<td>0</td>
</tr>
<tr>
<td>VNA</td>
<td><strong>0.7833</strong></td>
<td><strong>0.4943</strong></td>
<td>0.5209</td>
<td><strong>0.5389</strong></td>
<td><strong>3</strong></td>
</tr>
</tbody>
</table>

SAS-Rank with its mutation variations by 6 (3 MAP and 3 NDCG@10) values.

The reason for obtaining these performance results is that there is more exploration by various probability distribution random generators included in each Neighbourhood solution for the mutation process in VNA. Thus, VNA can jump from a local optimal solution to the global optimal one. On the contrary, including one probability distribution as a random generator for mutation step-sizes for each Neighbourhood solution causes less exploration for the near better optimal solution to the existing local optimal ones.

In table 8, a comparison between recent research studies in LTR problem with the proposed approaches [Oosterhuis, 2020, Vardashi et al., 2021]. The approaches used in this comparison are online and offline methods, Dueling Bandit Gradient Descent (DBGD), Multileave Gradient Descent (MGD), and Pairwise Differentiable Gradient Decent (PDGD) are online LTR methods used in
Table 8 A Comparison between the proposed methods with recent research studies

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Yahoo Set1</th>
<th>MSLR-WEB10K</th>
<th>MQ2008</th>
<th>MQ2007</th>
<th>Winning Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>VNA</td>
<td>0.7833</td>
<td>0.4943</td>
<td>0.5209</td>
<td>0.5389</td>
<td>3</td>
</tr>
<tr>
<td>DBGD (Linear)</td>
<td>0.684</td>
<td>0.331</td>
<td>0.683</td>
<td>0.483</td>
<td>1</td>
</tr>
<tr>
<td>DBGD (neural)</td>
<td>0.676</td>
<td>0.319</td>
<td>0.670</td>
<td>0.463</td>
<td>0</td>
</tr>
<tr>
<td>MGD (Linear)</td>
<td>0.714</td>
<td>0.333</td>
<td>0.690</td>
<td>0.494</td>
<td>0</td>
</tr>
<tr>
<td>Pairwise (Linear)</td>
<td>0.709</td>
<td>0.315</td>
<td>0.674</td>
<td>0.479</td>
<td>0</td>
</tr>
<tr>
<td>PDGD (Linear)</td>
<td>0.736</td>
<td>0.427</td>
<td>0.699</td>
<td>0.511</td>
<td>0</td>
</tr>
<tr>
<td>PDGD (neural)</td>
<td>0.743</td>
<td>0.430</td>
<td>0.699</td>
<td>0.509</td>
<td>0</td>
</tr>
<tr>
<td>LambdaMART</td>
<td>0.691</td>
<td>0.402</td>
<td>Nan</td>
<td>Nan</td>
<td>0</td>
</tr>
<tr>
<td>LambdaMART AC</td>
<td>0.763</td>
<td>0.477</td>
<td>Nan</td>
<td>Nan</td>
<td>0</td>
</tr>
<tr>
<td>LambdaMART MBC</td>
<td>0.767</td>
<td>0.474</td>
<td>Nan</td>
<td>Nan</td>
<td>0</td>
</tr>
<tr>
<td>DNN</td>
<td>0.716</td>
<td>0.415</td>
<td>Nan</td>
<td>Nan</td>
<td>0</td>
</tr>
<tr>
<td>DNN AC</td>
<td>0.746</td>
<td>0.448</td>
<td>Nan</td>
<td>Nan</td>
<td>0</td>
</tr>
<tr>
<td>DNN MBC</td>
<td>0.744</td>
<td>0.440</td>
<td>Nan</td>
<td>Nan</td>
<td>0</td>
</tr>
</tbody>
</table>

this comparison. The offline methods used in this comparison are two well-known algorithms with their correction of bias methods. These approaches are LambdaMART, LambdaMART with Affine correction (AC), LambdaMART with Mixture Based Correction (MBC), Deep Neural Networks (DNN), DNN with AC, and DNN with MBC. The details of these methods are demonstrated in [Oosterhuis, 2020, Vardasbi et al., 2021]. We included in this table, the results obtained from our methods in comparison with the best NDCG@10 results obtained by the researchers in previous research. From these results, we can find that VNA outperformed the other approaches in three datasets (Yahoo, MSLR-WEB10K, and MQ2007). While DBGD(linear model) outperformed other methods for MQ2008 dataset.

5 Conclusions

To sum up, this paper proposes a novel variable Neighbourhood method which is combining Variable Neighbourhood Search with Simulated Annealing (VNA). This proposed method is a Variable Neighbourhood Ascent used for each Neighbourhood mutation step-sizes with accepting (Move) bad solution when cooling temperature is reached. In the VNA procedure, a specific probability distribution is used for all mutations which are different from the other Neighbourhoods. More exploration for searching for optimal solutions was achieved by VNA using various probability distributions for mutation step-size in each Neighbourhood solution. This methodology helps in finding global optimum solutions without suffering from being stuck in local optimum solutions by less exploration for search solution space.

From the results obtained in table 2 to 8, we can conclude that VNA outperformed other LTR methods in recent and well-known LTR approaches for evolutionary and machine learning techniques. The Winning Rate columns have the number of the best performance for each method applied on the four datasets. Tables 2 to 8 results are demonstrated from the point-of-view of evolutionary computation researchers. Since the capabilities of the algorithms were measured by some of them from the meta-heuristic algorithm capabilities for evolving better solutions on the training data rather than predictive results on test data. Thus, we include in these tables the evolving and predictive results for training, validating, and testing datasets. From these results obtained, the total number of best performances using VNA is 16 (8 MAP and 8 NDCG@10)
values out of 24 evaluation metric values. On the other hand, the second method for obtaining the best performance is SAS-Rank with its mutation variations by 6 (3 MAP and 3 NDCG@10) values, while ESRank has the third best method performance by 2 (2 MAP) values.

From the other perspective, we compared the proposed methods with recent online and offline research studies by other researchers. Since SAS-Rank and EGS-Rank are two other methods introduced by us in recently published research [Ibrahim, 2022, Ibrahim and Younis, 2022]. The results of these recent research studies [Oosterhuis, 2020, Vardasbi et al., 2021] were compared to our proposed methods, and the results are provided in table 8. We can conclude that VNA outperformed the other approaches in three datasets (Yahoo, MSLR-WE10K, and MQ2007), while DBGD(linear model) outperformed other methods in MQ2008 dataset.

Declarations

Ethical Approval

There is no human or animal participation in this research.

Authors’ contributions

Dr. Osman wrote the code, reviewed the paper and wrote it.

Availability of data and material

The data are public data in the Microsoft Research website by Microsoft research team, while the source code is available in section 3.

Funding

Egyptian government supports open access subscription on hybrid Springer Journal.

Conflict of interest

The author declare that they have no conflict of interest.

References


Combining Variable Neighbourhood with Simulated Annealing for Learning to Rank Problem


Osman Ali Sadek Ibrahim. Exploration and exploitation methods using (1+1)-evolutionary algorithms for better ranking models, 2022. URL https://doi.org/10.21203/rs.3.rs-1968676/v1.


