An Optimized Rank Aggregation Approach for Mathematical Information Retrieval

Sourish Dhar, Assistant Professor, Department of Computer Science and Engineering, Assam University, India, Email:dharsourish@gmail.com

Sahinur Rahman Laskar, Guest faculty, Department of Computer Science and Engineering, Assam University, India, Email:sahinur07312@gmail.com

Sudipta Roy, Professor , Department of Computer Science and Engineering, Assam University, India, Email:sudipta.it@gmail.com

Abstract **Result merging or rank aggregation is one of the pivotal aspects of meta-search engines. The problem of result merging becomes substantial in the field of mathematical information retrieval systems. Math search engines is a new breed of information retrieval systems which takes a mathematical formula as query and provides relevant documents which contains the mathematical formula semantically or syntactically. This paper investigates various issues related to ranking of documents and proposes an optimized rank aggregation approach based on bio-inspired Grey Wolf Optimizer technique along with data fusion concept in the field of mathematical information retrieval systems. The proposed method is then implemented and compared with other state-of-the-art math aware search engines to deduce the fact that our approach outperforms other search engines.**

Keywords — ***Data fusion, Document ranking, Fitness Function, Grey Wolf Optimizer (GWO),Mathematical Information Retrieval (MIR),Result Merging.***

I. INTRODUCTION

Document ranking is one of the fundamental problems in information retrieval (IR)[1]. In the setting of search engines, based on a query, relevant documents are ordered in decreasing order of similarity score. Many ranking techniques were developed in the field of Information Retrieval, which can be categorized into two types: ranking creation and ranking aggregation. "Ranking creation is to create a ranking list of objects using the features of the object, while ranking aggregation is to create a ranking list of objects using multiple ranking list of the objects"[2]. The focus of this paper lies in rank aggregation , also known as result merging, in the domain of mathematical information retrieval (MIR). As traditional text retrieval systems are not suitable for handling mathematical expressions since mathematical expression constitute a range of symbols to complex structures without disregarding the order [3]. This very fact motivated us to work in the direction of creating a rank aggregation scheme in the domain of mathematical search engines or MIR.

In this paper, we investigated ranking result of existing search engines available in the domain of math information retrieval (MIR) and proposed an optimized rank aggregation scheme based on Grey Wolf Optimizer (GWO) along with data fusion techniques to handle critical issues presented by existing mathematical search engines.

The remainder of the paper is organized as follows. In Section II, a brief overview of related work is presented. The problem description is elaborated Section III. The proposed fitness function is explained in Section IV and in section V we explain the proposed algorithm. The experimental result and its interpretation are provided in section VI and finally, concluding remark is presented in section VII.

II. RELATED WORK

In this section, we briefly review and comment on the different aspects of our study and related contribution.

A. Result Merging

Result merging or rank aggregation is the core of meta search engine[4]. In this process an information need queried by the user in a meta-search engine gets a merge result which are ranked in a single rank list . In the background, meta search engine reprocess the user query and sends it to underline appropriate set of search engines through which results are retrieved, merged and ranked in a single ranked list. The algorithms for result merging can be classified into four categories [5]:

1. Round Robin based method: In this method, in each round and in a certain order one result is taken from the result list of underline search engine.
2. Similarity conversion-based method: The local ranks of the component search engine are converted into similarities to apply similarity-based merging techniques.
3. Voting based method: Here each component search engine is modeled as a voter and each result is

 DOI : 10.18231/2454-9150.2018.0639

modeled as a candidate in an election. The basic idea is to build a consensus among voters.

4. Machine learning based method: Based on a training data the machine learns the merged result list with which it tries to predict the ranking of the testing data.

Many ranking techniques were developed in the field of Information Retrieval (IR). In [6], Collection Retrieval Interface (CORI) merging algorithm was used to normalize the document scores retrieved from each component search engine. In [7], another merging technique was used to calculate the merging score based on the number of retrieved documents. It actually scores the search engine based on the proportion of retrieved results. In [8], a semi supervised learning algorithm was employed which is based on overlap measures between underlying search engines and centralized search engine from which a mapping is generated between the corresponding search engine score using a learning function. In [9], a learning-based approach was used for search engine selection and Ordered Weighted Averaging (OWA) based merging approach was used which is based on fuzzy set theory.

B. *Data Fusion*

Data fusion schemes find its application ranging from sensor networks to information retrieval.

Five combination function for combining scores was described by Fox and Shaw,1994[10,11].They are :

CombMIN = Minimum of Individual Similarities

CombMAX = Maximum of Individual Similarities

CombSUM = Summation of Individual Similarities

CombANZ = CombSUM ÷ Number of non zero Similarities

CombMNZ = Comb SUM × Number of non zero Similarities.

As per [12,13,14] fusion functions which are different from Comb- functions with respect to the generation of answer sets. Based on relevance score , documents are assigned ranks using fusion functions. Data fusion functions are more feasible for optimizing search results in the mathematical information retrieval (MIR). In this regard, a single list is created by merging the documents using the Rank Position method[12,15] as given below:

(1)

$$r(d_i) = \frac{1}{\sum_j 1/_{position(d_{i,j})}}$$

Here, the computation of the rank score of document i, using the position information of this document across all component search engines (j = 1. . .m).

The method proposed in [12], is a multilayer technique to maximize precision and improve the retrieval performance that satisfies the user needs. They suggested by invoking their method and integrating other techniques might lead to better performance. Data fusion concept was also used in [15], for automatic ranking of retrieval systems. They merged the retrieval results of multiple systems using a hybrid of Rank position method, Borda count method and Condorcet method. Hence obtained top-ranked documents in the merged result as the ‘‘(pseudo) relevant documents,’’ and employed these documents to re-rank in the retrieval systems. . In both of the approaches mentioned above, relevance judgments was not considered as a feature which we consider a crucial factor for optimizing result merging process using data fusion technique.

C. *Existing Math Information Retrieval Systems(MIR)*

Unlike text, the structure of Mathematical Expression (ME) and formulae may be multidimensional. Majority of math aware search engines or MIR systems handles this aspect by using two approaches of indexing namely text-based or tree based.

In text-based approach, a mathematical formula markup is converted into plain text string. Then the traditional text retrieval schemes are implemented using existing information retrieval tools like Lucene. This conversion itself presents another set of challenges like retaining structure information of MEs into the string, resolving notational ambiguity and normalization of the string [16]

Canonical form for normalized operands was proposed by Miller et. al. [17,18] after converting all non-alphanumeric symbols in LATEX into alphanumeric symbols. On the same grounds, later Misutka et al. [19] optimized the design by incorporating an additional normalization of variables and constants. But in all these methods the structural information of mathematical expressions cannot be recovered as it gets lost during the process. Moreover, Miner et al. [20] proposed n-gram indexing for substructure matching, setting n to five. Again this resulted in fewer nodes indexing causing ineffective index for complex structures.

In tree based approach, mostly symbol layout tree(SLT) or operator tree(OPT) are employed to index formulae. Attributes of tree structures like sub-expression or path are extracted as index terms. Indexing all substructures of a formula can lead to high recall but suffers from index size growth. To address the issue of indexing the sub-structures of semantic formula a substitution tree indexing technique was proposed by Sojka et. al. [21] and Kohlhase et al. [22]. Originally substitution tree indexing was proposed by Graf [23] for theorem provers. Similarly,it was also used by Schellenberg et. al.[24] for indexing layout presentation of formulae. All these approaches although improves precision but offer inferior recall.

As per our study, we are not aware of any mathematical meta search engine developed yet and the only contribution made for optimizing rank result in math similarity search is provided in [25]. Their work was based on correlation measurement whereas our work is based on human relevance or in other words relevance judgement.

We have chosen MIaS[26] and WikiMirs[16,27] for the purpose of comparison due to the availability of human relevance judgement.

 DOI : 10.18231/2454-9150.2018.0639

D. Grey Wolf Optimizer(GWO)

Originally, Grey Wolf Optimizer (GWO) was coined by S. Mirjalilli in year 2014 [29], which mimics the leadership hierarchy of wolves and their hunting behavior. It is a simulation to recognize optimum unique hunting agent among pack of grey wolves and searching prey characteristics. Four levels of social hierarchy of grey wolves exists in GWO namely

1. First level : It is also considered as the leader of the pack and known as α (best hunting agent) . It has the responsibility to take decision, manage and control the whole pack.
2. Second level: Also known as β (second best hunting agent) which assist the first level for taking decisions.
3. Third level : These are the subordinates who are assigned tasks to execute. They also help the first level and second level in various tasks like hunting, boundary watching etc. These are also known as δ.
4. Fourth level : This is the lowest level and known as Ω wolves. They have to submit to all the other levels. The last level Ω wolves followed by δ, responsible for maintaining the safety and integrity in the wolf pack.

The distance α,β and δ wolves i.e. D_α , D_β and D_δ to each of the remaining wolves ($\vec{X}$) are calculated using equation (2)[30] using which the effect of α, β and δ wolves on the prey viz. $\overrightarrow{X_1}$, $\overrightarrow{X_2}$, $\overrightarrow{X_3}$ can be calculated as represented in equation (3)

$$\{\overrightarrow{D_\alpha}=|\overrightarrow{C_1}.\overrightarrow{X_\alpha}-\vec{X}|, \overrightarrow{D_\beta}=|\overrightarrow{C_2}.\overrightarrow{X_\beta}-\vec{X}|, \overrightarrow{D_\delta}=|\overrightarrow{C_3}.\overrightarrow{X_\delta}-\vec{X}|\} \quad (2)$$

$$\{\overrightarrow{X_1}=\overrightarrow{X_\alpha}-\overrightarrow{A_1}.\overrightarrow{D_\alpha}, \overrightarrow{X_2}=\overrightarrow{X_\beta}-\overrightarrow{A_2}.\overrightarrow{D_\beta} , \overrightarrow{X_3}=\overrightarrow{X_\delta}-\overrightarrow{A_3}.\overrightarrow{D_\delta}\} \quad (3)$$

$$\vec{A}=2\overrightarrow{a}.\overrightarrow{r_1}-\overrightarrow{a}, \vec{C}=2.\overrightarrow{r_2} \quad (4)$$

$$\vec{X}(t+1)=\left(\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}\right)/3 \quad (5)$$

The values of controlling parameters of the algorithm which are α, A and C are calculated using Equation (4). Here, $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$ are the random vectors in the range of [0,1]. These vectors make wolves able to reach at any point between the prey and the wolf. Vector $\overrightarrow{r_1}$ is involved in controlling activity of the GWO algorithm and used in calculating $\vec{A}$. The component values of $\overrightarrow{a}$ vector decreases linearly from 2 to 0 over the courses of iterations. $\vec{C}$ helps in putting some extra weight on the prey to make it difficult for the wolves to find it. Finally, all other wolves update their positions $\vec{X}(t+1)$ using Equation (5) [29,30,31].

We propose a bio-inspired optimization algorithm in the domain of mathematical information retrieval i.e. Grey Wolf Optimizer (GWO) to model the problem of rank aggregation using data fusion techniques based on relevance judgement.

III. Problem Description

We have considered a label results for each system ranging from {1,2,..,5}. Here, label 1 represents the results which are not relevant to the query and higher score denotes results more relevant to the query. Based on our observation and study, the following challenges have been encountered. Here SE1,SE2 and SE3 represent component search engines. The following problems are encountered based on result merging from different component search engines.

A. Missing Rank Position

Different search engines provide different rank results. But many documents may have same relevance judgement, yet they are not ranked by one or more search engines. So, it is necessary to rank those documents which are not captured by the search engines.

Doc ID	Relevance judgement	Rank position@10		
		SE1	SE2	SE3
1	4	4	4	
2	4			5
3	4	9	9	

Table 1 Missing rank position problem having similar relevance judgments

Consider Table 1, here Doc ID 1 has relevance judgement of 4 but it is not captured by all the search engines as it is observed that search engine SE3 did not rank it at all. So, it can be concluded that SE3 could not rank it properly. Similarly, in-spite of Doc Id 2's relevance judgement of 4, it is also not ranked by search engines SE1 and SE2. Hence, these two search engines also could not provide document ranking efficiently.

In a nutshell, it is observed that all these documents are not present in the common ranking list of all the search engines. Therefore, considering the above fact it is necessary to rank these documents efficiently.

B. Different Rank Position (Same Relevance judgement)

Different search engines may provide different rank positions to the documents having same relevance judgments. Different documents having different rank positions in different search engines with same relevance judgments itself poses a challenge on the efficiency of the search engines. The problem is illustrated in Table 2.

Table 2 Different rank position problem having similar relevance judgement

Doc ID	Relevance judgment	Rank position@10		
		SE1	SE2	SE3
1	4	4	2	6
2	4	2	4	5
3	4	9	1	3

C. Different Rank Position (Different Relevance judgement)

Different search engines may provide different rank positions for different relevance judgement. Documents

 DOI : 10.18231/2454-9150.2018.0639

having higher relevance judgements might get lower ranking positions in different search engines, which also becomes a factor to the efficiency of the search engines. The problem is illustrated in Table 3.

Table 3 Different rank position problem having dissimilar relevance judgement

Doc ID	Relevance judgement	Rank position@10		
		SE1	SE2	SE3
1	4	4	1	2
2	3	2	4	1
3	2	1	2	3

D. Hybrid

The fourth issue arises as a combination of the above three points which is hybrid problem shown in Table 4.In real world, the aforementioned problems are evident as most of the search engines results differ due to their underlying document representations, indexing schemes, similarity matching and the datasets.

Table 4: Hybrid problem

Doc ID	Relevance judgement	Rank position@10		
		SE1	SE2	SE3
1	4	1		6
2	4	2	1	1
3	3	3		5
4	2	4	2	3

IV. PROPOSED FITNESS FUNCTION

As discussed earlier, the Grey wolf optimizer technique along with data fusion concept, is proposed here in which we draw the analogy of all the problems discussed in previous section. In this method, all the ranked documents are considered as Grey wolves and all the ranked document have different ranked position in different search engines like all Grey wolves have different position in different dimensions.

In Grey Wolf Optimization scenario, the fitness value of every wolves is calculated in every dimension based on its position vector from target position i.e. prey position. Then based on fitness value, wolf position is updated. While calculation of fitness value, the position of the prey is uncertain, hence in every dimension a random position value is generated.

Similarly, we considered fitness value of every document in every component search engine. But in our context by only considering position of ranked document, it is not possible to calculate fitness value of each document as every document may have different relevance judgments along with different ranked position.

For instance, consider the following example given in Table 5.

Table 5: Different relevance judgments along with different ranked position

Doc Id	Relevance judgement	SE1
1	4	2
2	3	4
3	2	1

Here, Doc ID 3 have top position in search engine SE1, but relevance judgment is less than Doc ID 1. So, to calculate fitness value document it is necessary to consider relevance judgment of every document. Also, only considering relevance judgment as fitness value is not justified to select top position because, there might be possible more than one has same relevance judgments as shown in Table 6.

Table 6: Same relevance judgments for multiple documents

Doc ID	Relevance judgement	SE1
1	4	2
2	4	4
3	4	1

Hence by considering both parameter i.e. relevance judgment and ranked position, fitness value of the document can be calculated using the equation(6) as given below:

$$FV_{id}^{j} = K \times IFV_{id} + \frac{1}{PS_{id}^{j}} \quad (6)$$

Where ,

FV_{id}^{j} = Fitness value of document id (id=1,2…n) corresponding search engine j(j=1,2….m).

IFV_{id}=Initial Fitness Value of document i (Assumed relevance judgement of document i)

PS_{id}^{j} =Position score of document id in the search engine j.

$K$is a flag variable is used to recognize whether the document is present or missing, K=1 if document exists , otherwise 0.

The procedure is further illustrated below. Consider the following scenario shown in Table 7.

Table 7: Fitness Value Calculation

Doc ID	Relevance judgement	SE1	FV_{id}^{1}
1	3	1	3
2	4	2	4.5
3	4	3	4.3
4	2	α(infinity)	0

To omit document which are not ranked, we consider its position as α (infinity) which is different from α-solutions as the output of GWO.

The main advantage of this model is that it guarantees to provide unique fitness value of the document. In simple words, every document has different fitness value because

 DOI : 10.18231/2454-9150.2018.0639

each document has unique ranked position in the considering current search engine.

In Grey Wolf Optimizer scenario, after calculation fitness value of hunting agents (wolves) next task is to find out best hunting agent α, second best hunting agent β, third best hunting agent δ and the rest are known as ω which are following delta. With this connection, we traverse all the search engines corresponding to every document. As a result, for every search engine we have different fitness value of every document. So, different search engines will have different *α, β and δ* and might lead to an ambiguous condition which results in having same fitness value of more than one document. Hence, by only considering total fitness value it is not possible to recognize unique optimum solutions.To elaborate the problem , consider the following Table 8.

Table 8: Calculation of $\sum FV_{id}^{j}$

Doc ID	SE1	SE2	SE3	$\sum FV_{id}^{j}$
1	4.5 (α)	1.5 (α)	2.5 (β)	**8.5 (same)**
2	2.5(β)	4.5 (β)	0	7.5
3	0	0	1.5 (δ)	1.5
4	1.5 (δ)	2.5 (δ)	4.5 (α)	**8.5(same)**

Now to solve this problem using data fusion concept, we extended (6) into our proposed model given in (7).

$$FFV_{id} = K \times \sum_{j=1}^{m} FV_{id}^{j} + \frac{1}{RPS_{id}} \qquad (7)$$

Where,

FFV_{id} =Final fitness value of the considering ranked document.

$k = 1$, since all the ranked document are considered.

FV_{id}^{j} =Fitness value of the document id in the considering search engine j.

RPS_{id} =Ranked position score of the document id.

To further illustrate consider the following scenario:

Table 9: A running example

Doc ID	Relevance Judgement	SE1	SE2	SE3	$\sum FV_{id}^{j}$	RPS_{id}
1	4	2	4	α(infinity)	8.75	1.33
2	3	1	α(infinity)	2	7.5	0.66
3	2	α(infinity)	3	1	5.33	0.75

Using equation 1 we get,

$$RPS_1 = \frac{1}{(\frac{1}{2}+\frac{1}{4}+0)} = \frac{1}{0.5+0.25+0} = 1.33$$

Similarly,

RPS_2=0.66

RPS_3=0.75

Now using equation 7 we get,

$$FFV_1 = 1 \times 8.75 + \frac{1}{1.33} = 9.50$$

$$FFV_2 = 1 \times 7.5 + \frac{1}{0.66} = 9.01$$

$$FFV_3 = 1 \times 5.33 + \frac{1}{0.75} = 6.66$$

and finally, we get our desired results tabulated in Table 10.

Table 10: Final Fitness Value (FFV)

Doc ID	RPS_{id}	$\sum FV_{id}^{j}$	FFV_{id}
1	1.33	8.75	9.50 (α -solution)
2	0.66	7.5	9.01 (β -solution)
3	0.75	5.33	6.66 (δ-solution)

V. PROPOSED ALGORITHM

We have considered five different search engines based on different indexing scheme and similarity metrics. In our earlier work [32] we have created a meta-search engine for scientific document retrieval based on different indexing schemes namely Pattern Based Trie (PBT), Lucene [33] with Structure Encoded String (SES) , Lucene(Vector Space Model) .. We have also considered two other state-of-the-art math-aware search engine namely MIaS , Wikimirs1 as discussed earlier and obtained ranked documents corresponding to different standard queries.

A. *Algorithm*

Input: Let a set S= {Uq, D_i}, where U_q is the user query and D_i = {D_1, D_2, D_3… D_n} is list of document return by the underlying search engine SE_j (j=1,2…m). Since we considered five search engines, so value of m is set to 5.

Output: Single rank list Dr{r =1,2,3….n} of document after merging the result, where D_1=D_α ,D_2=D_β ,D_3=D_δ.

Method:

Begin

Step 1: Generate the random population of documents in the search space along with their rank position and relevance judgments.

Step 2: Calculate the Rank position score (RPS) of each individual of population document across all component search engines by considering equation (1) .

$$r(d_i) = \frac{1}{\sum_j 1/position(d_{ij})} \qquad (1)$$

Where,

$r(d_i)$=**RPS$_i$** is the rank position score of document i (i=1,2….n) across all the search engine SE_j (j=1,2…m).

Step 3: Check whether end of results is encountered for the current search engine SE_j ?

 DOI : 10.18231/2454-9150.2018.0639

If True goto **Step 7**; else goto **Step 4**.

Step 4: Calculate fitness value (FV) of each individual corresponding to the current search engine SE_j, using equation (6).

$$FV_{id}^{j} = K \times IFV_{id} + \frac{1}{PS_{id}^{j}} \qquad (6)$$

Step 5: Update the rank position of the individuals based on fitness value (FV).

Step 6: Find the current value of α(alpha), β(beta) and δ(delta) using individuals position with respect to current search engine SEj and repeat the process from **Step 3** to **Step 6** for all participant search engines.

Step 7: Compute the final fitness value (FFV) of all the ranked individuals by using (7).

$$FFV_{id} = K \times \sum_{j=1}^{m} FV_{id}^{j} + \frac{1}{RPS_{id}} \qquad (7)$$

Step 8: Finally obtain the optimum solution by finding the values of α, β and δ among n individuals.

End

B. Flowchart

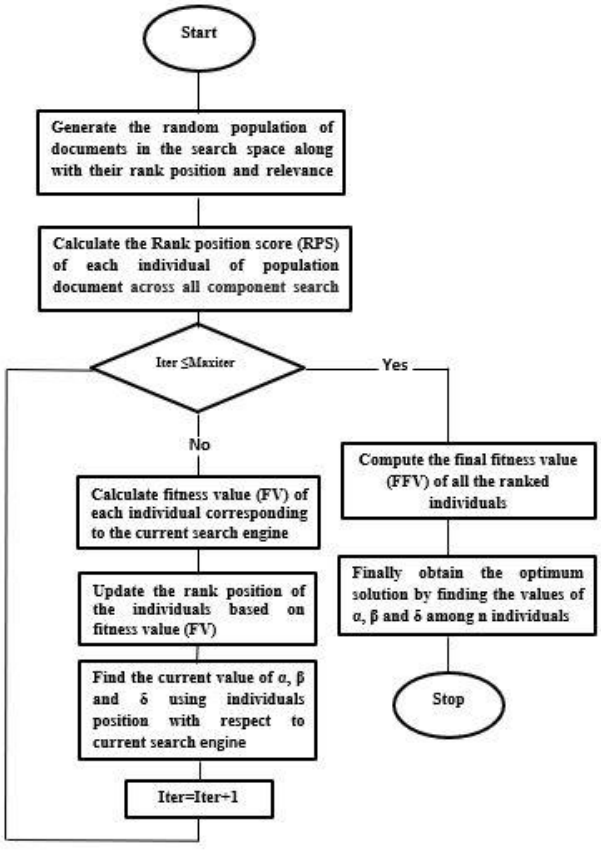

Fig1. Flowchart of the proposed algorithm

VI. Experimental Results

We evaluated our system using NTCIR-12 Math-IR[1] task (http://ntcir-math.nii.ac.jp/.) from which Wikipedia Corpus was downloaded that contain mathematical formulas written for normal users.

(i) Wikipedia Corpus contains 319,689 articles from English Wikipedia converted into simpler XHTML format with images removed. There are around 31,839 MathTag articles which is approximately 10% of the collection approximately and 287,850 Text articles which contributes 90% of the collection approximately. There are around590,000 formulas in this corpus encoded using presentation and content MathML. The size of this corpus having uncompressed documents is 5.15GB .

(ii) Query Set Description The query set was downloaded from [34] along with necessary relevance judgments. The query set is presented in JSON format, which is composed of with approximately 100 queries. Each query contains a query string in LATEX along with list of labels containing the URL and its score.

A. Evaluation Measure

To evaluate the performance of results, Precision and Discounted Cumulative Gain (DCG) @10 [35], are calculated respectively, over 25 random queries chosen from the query set.

Precision: It measures the exactness of the retrieval process. If the actual set of relevant document is denoted by I and the retrieved set of document is denoted by O, then the precision is given by:

$$Precision = \frac{|I \cap O|}{|O|} \qquad (8)$$

Discounted Cumulative Gain(DCG): DCG measures the usefulness, or gain, of a document based on its position in the result list. DCG of the top-k retrieved results can be calculated using equation (9) as shown below:

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)} \qquad (9)$$

B. Result

We compared the ranked result of our approach with the ranked result of other search engines namely Pattern Based Trie , Lucene with Structure Encoded String (SES), ,Lucene(Vector Space Model) , MIaS and Wikimirs1 in terms of Precision@10, which is shown in Table 11 and Fig.2.Similarly, compared ranked result of our approach with other search engines in terms of DCG@10, is shown in Table 12 and Fig.3.

In the Table 11 and Fig.2, it is observed that most of the results based on queries retrieved through our approach shows better precision value than that of other search engines.

Few queries like Q9 and Q14 has a better precision value in in MIaS and for Q25, Wikimirs1 shows better precision value. However, in case of our approach, mean value of precision@10 value over 25 queries is 0.76 which is better result in comparison of other search engines: Pattern Based Trie (0.664), Lucene (SES) (0.664), Lucene (VSM) (0.316), MIaS (0.34) and wikimirs1(0.124).

Similarly, comparison is done in terms DCG@10 in Table 12 and Fig.3. It is observed that Q3,Q13 and Q19,the DCG remains the same and consistent.For the rest of the queries, our approach provides better or optimum results than considering the results of all the search engines . We estimated mean value of DCG @10 for the set of 25 queries, in case of our approach it shows 13.9864 which is better than pattern Based Trie (12.7064), Lucene (SES)

[1] http://ntcir-math.nii.ac.jp/.

 DOI : 10.18231/2454-9150.2018.0639

(12.6052),Lucene (VSM) (7.9268), MIaS(3.2852) and Wikimirs1(3.3824)

Table 11: Precision@10 Comparison PBT,SES,VSM,MIaS,Wikimirs1 Vs Our Approach

Query Id	Mathematical Notation/LaTeX Query	Pattern Based Trie (PBT)	Lucene (SES)	Lucene (VSM)	MIaS	Wikimirs1	Our Approach
		Precision @10					
Q1	(¬q∨¬q)↔(p→¬q)	0.5	0.5	0.2	0.3	0	0.6
Q2	(a-b)^2 = a^2+b^2-2ab	0.9	0.9	0	0.2	0	0.9
Q3	(x⊕y)=(x+7)(x+4)−6k	0.1	0.1	0.1	0.4	0	0.1
Q4	\int \frac{\sin x}{x}dx	0.6	0.6	0.6	0.5	1	0.7
Q5	\neg(p \vee q) \vee (\neg p \wedge q) = \neg p	0.6	0.6	0.2	0.2	0	0.7
Q6	a \equiv b \pmod n	0.1	0.1	0.1	0.5	0.1	0.6
Q7	{6.7}^n - {2.3}^n	0.7	0.7	0.1	0	0/5	0.8
Q8	((p→(q→r))→((p→q)→(p→r)))	0.8	0.8	0.7	0	0.1	0.9
Q9	(p-1)! \equiv -1 (\mod {p})	0.1	0.1	0.1	0.3	0	0.1
Q10	(a^2+b^2+c^2)^2 = 2(a^4+b^4+c^4)	0.9	0.9	0.1	0.3	0	0.9
Q11	(n-m) \mid (n^k-m^k)	0.8	0.8	0.7	0.8	0	0.9
Q12	(p \leftrightarrow q)	0.4	0.4	0.1	0.1	0.3	1
Q13	(p \vee q) \wedge (p \to r) \wedge (p \to r) \to r	0.9	0.9	0.3	0	0	0.9
Q14	(x+y)\times\frac{a}{b}	0.1	0.1	0.1	0.3	0	0.1
Q15	(x^2+1)^2	0.9	0.9	0.8	0.4	0.3	1
Q16	(z + y + x)^2	0.9	0.9	0.9	0.4	0	0.9
Q17	1+\tan^2x	0.9	0.9	0.1	0.1	0	0.9
Q18	1+x+x^2+x^3+\ldots \:	0.9	0.9	0.7	0.5	0	0.9
Q19	1+x+x^2+x^3=y	0.9	0.9	0.1	0.2	0	0.9
Q20	1.1!+\dots+n.n! = (n+1)!	0.8	0.8	0.1	0.4	0	0.9
Q21	10 \sin^2(\theta) - 13 \sin(\theta) - 3 = 0	0.9	0.9	0.9	0.9	0	1
Q22	2+4+\dots+2n=n(n+1)	0.9	0.9	0.6	0.3	0	1
Q23	3x^2 + 2x	0.9	0.9	0.1	0.5	0	0.9
Q24	7x + 5y = 3 \pmod 4	0.6	0.6	0.1	0.4	0	0.7
Q25	F_{n}^{2}-(F_{n+1})(F_{n-1})=(-1)^{n-1}	0.5	0.5	0.1	0.5	0.8	0.7
	Mean	**0.664**	**0.664**	**0.316**	**0.34**	**0.124**	**0.76**

Query Id	Mathematical Notation/LaTeX Query	Pattern Based Trie (PBT)	Lucene (SES)	Lucene (VSM)	MIaS	Wikimirs1	Our Approach
		DCG @10					
Q1	(¬q∨¬q)↔(p→¬q)	12.29	12.29	9.17	3.74	7.95	14.14
Q2	(a-b)^2 = a^2+b^2-2ab	14.49	14.49	14.48	1.51	0	15.47
Q3	(x⊕y)=(x+7)(x+4)−6k	9.08	9.08	7.76	1.3	0	9.08

 DOI : 10.18231/2454-9150.2018.0639

Q4	\int \frac{\sin x}{x}dx	15.24	15.23	13	3.31	4.54	16.33
Q5	\neg(p \vee q) \vee (\neg p \wedge q) = \neg p	12.68	12.68	9.17	2.15	0	14.68
Q6	a \equiv b \pmod n	5.54	6.89	6.13	17.73	15	10.04
Q7	{6.7}^n - {2.3}^n	11.83	11.7	3.89	0	3.94	12.7
Q8	((p→(q→r))→((p→q)→(p→r)))	12.45	12.45	10.57	0	15	14.28
Q9	(p-1)! \equiv -1 (\mod {p})	5.65	5.65	4.54	0	0	6.65
Q10	(a^2+b^2+c^2)^2 = 2(a^4+b^4+c^4)	14.59	14.58	4.54	2.42	0	14.97
Q11	(n-m) \mid (n^k-m^k)	13.17	12.78	7.25	3.82	0	14.31
Q12	(p \leftrightarrow q)	12.54	16.74	9.17	4.51	31.96	19.92
Q13	(p \vee q) \wedge (p \to r) \wedge (p \to r) \to r	15.26	15.26	9.12	0	0	15.26
Q14	(x+y)\times\frac{a}{b}	6.85	6.85	8.2	9.96	0	10.07
Q15	(x^2+1)^2	16.13	13.97	4.85	2.84	2.13	16.13
Q16	(z + y + x)^2	13.2	13.2	6.94	4.34	0	13.34
Q17	1+\tan^2x	16.62	16.61	8.29	0.33	0	17.19
Q18	1+x+x^2+x^3+\ldots \:	18.12	13.15	4.87	2.66	0	18.32
Q19	1+x+x^2+x^3=y	13.63	13.63	7.2	0.61	0	13.63
Q20	1.1!+\dots+n.n! = (n+1)!	12.68	12.68	4.54	4.21	0	13.04
Q21	10 \sin^2(\theta) - 13 \sin(\theta) - 3 = 0	13.63	13.63	14.05	6.58	0	13.63
Q22	2+4+\dots+2n=n(n+1)	14.58	14.57	11.49	1.81	0	15.35
Q23	3x^2 + 2x	15.52	15.13	5.89	2.19	0	15.37
Q24	7x + 5y = 3 \pmod 4	9.88	9.88	5.92	2.26	0	13.37
Q25	F_{n}^{2}-(F_{n+1})(F_{n-1})=(-1)^{n-1}	12.01	12.01	7.14	3.85	4.04	12.39
	Mean	**12.7064**	**12.6052**	**7.9268**	**3.2852**	**3.3824**	**13.9864**

Table 12: DCG@10 Comparison PBT,SES,,VSM,MIaS,Wikimirs1 Vs Our Approach

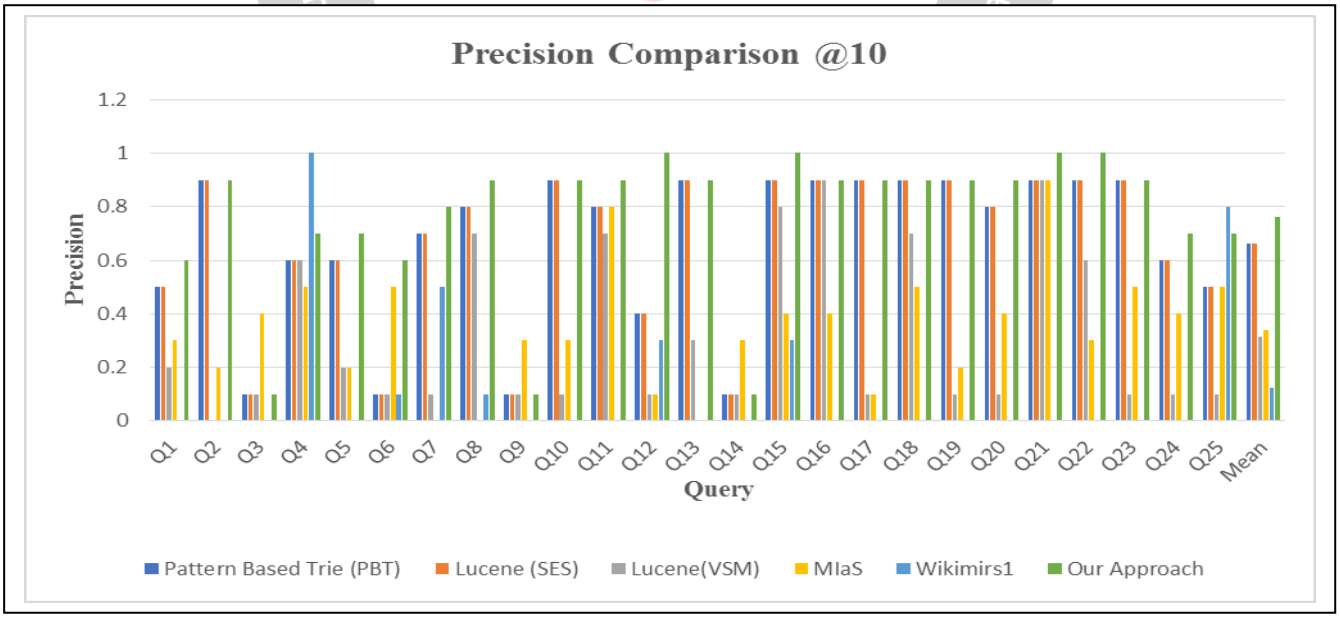

Fig2. Precision @10 comparison: Pattern Based Trie (PBT),Lucene (SES),Lucene(VSM) MIaS and Wikimirs1 vs Our approach

 DOI : 10.18231/2454-9150.2018.0639

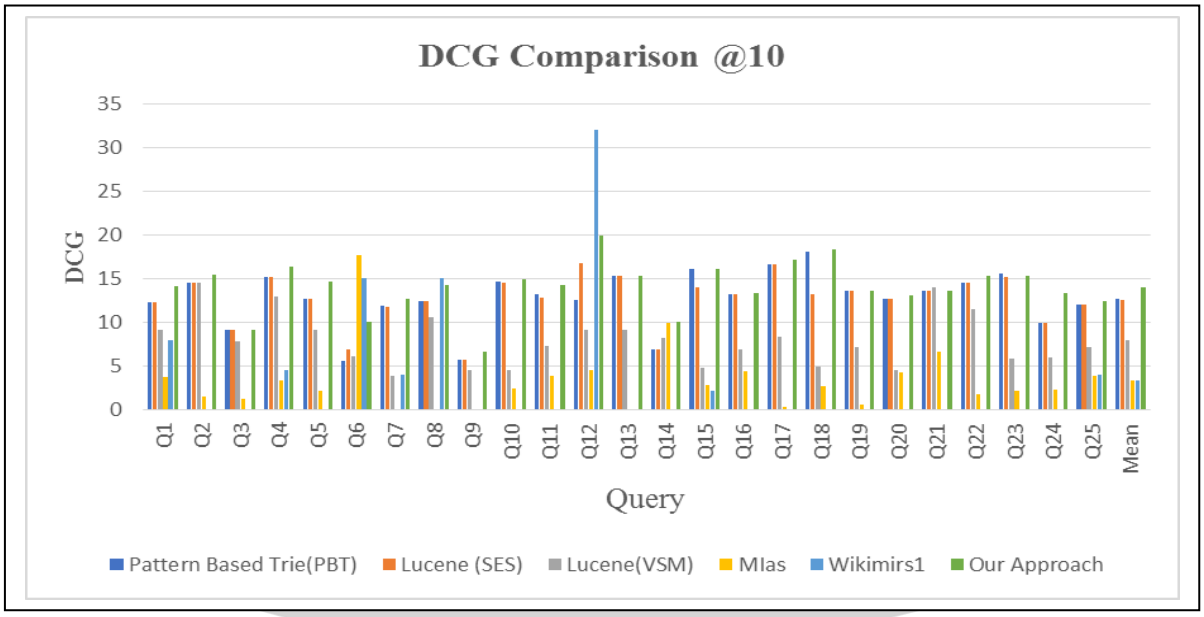

Fig3. DCG @10 comparison: Pattern Based Trie (PBT),Lucene (SES),Lucene(VSM) MIaS and Wikimirs1 vs Our approach

VII. Conclusion

This effort was focused towards issue of result merging in meta-search engine in the domain of mathematical information retrieval. We proposed a heuristic approach based on the principles of Grey Wolf Optimizer and data fusion schemes. As per our approach, we obtained the fitness value of the retrieved results across different search engines depending on the rank position and relevance judgments of the results. With our approach we deduced that the document having the best fitness value moves towards the top of the single ranked list based on rank position method. Our proposed model also takes into account of the missing results.

In this approach, all the results retrieved from each component search engine are considered as grey wolves. To find the values of α, β, δ –solutions we employed rank position method which is data fusion technique and proposed a heuristic approach of including relevance judgement in the model. Other candidate solutions i.e. Ω wolves are updated based on the position of α, β, δ depending on the final fitness score. The process is iteratively called for all the solutions in each component search engine until the optimum α, β, δ –solutions are obtained and a single rank list is created.

The comparison was done with other state-of-the art math aware search engines on the basis of precision and discounted cumulative gain to measure the usefulness of our results. In summary, from the comparative analysis , our approach facilitates result which yields an optimized result that performs better than other search engines.

We conclude that our results outperformed or optimized the rank list results in comparison to other systems. In future, other optimization techniques like gravitational algorithm, blue whale optimization techniques may be exploredl in context of the problem and other combination functions may be used to further enhance. Weighting schemes of the documents may also be exploited in this construction. The field of MIR is still young and has a vast scope of implementing new theories and principles in this engaging field to create a mature system.

References

[1] Andrei Broder and Monika Henzinger. “Algorithmic Aspects of Information Retrieval on the Web”. In: Handbook of Massive Data Sets. Ed. By James Abello, Panos M. Pardalos, and Mauricio G. C. Resende. Boston, MA: Springer US, (2002), pp. 3–23. ISBN: 978-1-4615-0005-6. DOI: 10 . 1007/978-1-4615-0005-6_1. URL: https://doi.org/10.1007/978-1-4615-0005-6_1.

[2] Hang Li. “Learning to Rank for Information Retrieval and Natural Language Processing”. Morgan and Claypool Publishers, (2011). ISBN: 1608457079, 9781608457076.

[3]Amarnath Pathak et al. “MathIRs: Retrieval System for Scientific Documents”. In: Computación y Sistemas 21.2(2017). URL:http://www.cys.cic.ipn.mx/ojs/index.php/CyS/article/view/2743.

[4] J. Kumar, R. Kumar, and M. Dixit. “Result merging in meta-search engine using genetic algorithm”. In: International Conference on Computing, Communication Automation. (2015), pp. 299–303. DOI: 10.1109/CCAA. 2015.7148393.

[5] Weiyi Meng and Clement Yu. “Advanced Metasearch Engine Technology”. Ed. by M. Tamer Ozsu. Morgan & Claypool Publishers, (2010). ISBN: 1608451925, 9781608451920.

[6] Henrik Nottelmann and Norbert Fuhr. “Combining CORI and the Decision Theoretic Approach for Advanced Resource Selection”. In: Advances in Information Retrieval. Ed. by Sharon McDonald and John Tait. Berlin, Heidelberg: Springer Berlin Heidelberg, (2004), pp. 138–153. ISBN: 978-3-540-24752-4.

[7] Yves Rasolofo, Faïza Abbaci, and Jacques Savoy. “Approaches to Collection Selection and Results Merging for Distributed Information Retrieval”. In: Proceedings of the Tenth International Conference on Information and KnowledgeManagement. CIKM’01. Atlanta, Georgia, USA:

 DOI : 10.18231/2454-9150.2018.0639

ACM,(2001), pp. 191–198. ISBN: 1-58113-436-3. DOI:10.1145/502585.502618. URL: http://doi.acm.org/10.1145/502585.502618.

[8] Luo Si and Jamie Callan. "A semisupervised learning method to merge search engine results". In: ACM Trans. Inf. Syst 21.4 (2003), pp. 457–491.

[9] R Kumar and A K Giri. "Learning based approach for search engine selection in meta-search engine". In: IJEMR. India 3 (2013), pp. 82–88.

[10] Joseph A. Shaw et al. "Combination of Multiple Searches". In: The Second Text REtrieval Conference TREC-2. (1994), pp. 243–252.

[11] Javed A. Aslam and Mark Montague. "Models for Metasearch". In:Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '01. New Or-leans, Louisiana, USA: ACM, (2001), pp. 276–284. ISBN: 1-58113-331-6. DOI: 10.1145/383952.384007.URL: http://doi.acm.org/10.1145/383952.384007.

[12] S. V. Saravanan M. Sivaram R. Abirama Krishnan B. Dhivakar. "Statistical Score Calculation of Information Retrieval Systems using Data Fusion Technique". In: Scientific & Academic Publishing 2.2 (2012), pp. 43–45.

[13] Shengli Wu. "Data Fusion in Information Retrieval". Springer Publishing Company, Incorporated, (2012). ISBN: 3642288650, 9783642288654.

[14] L. Valet, G. Mauris, and P. Bolon. "A statistical overview of recent literature in information fusion". In: IEEE Aerospace and Electronic Sys-tems Magazine 16.3 (2001), pp. 7–14. ISSN: 0885-8985. DOI: 10.1109/62.911315.

[15] Rabia Nuray and Fazli Can. "Automatic Ranking of Information Retrieval Systems Using Data Fusion". In: Inf. Process. Manage. 42.3 (2006), pp. 595–614. ISSN: 0306-4573. DOI: 10.1016/j.ipm.2005.03.023. URL: http://dx.doi.org/10.1016/j.ipm.2005.03.023.

[16] Y. Wang, L. Gao, S. Wang, Z. Tang, X. Liu, K. Yuan, "Wikimirs 3.0: A hybrid mir system based on the context, structure and importance of formulae in a document", Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries pp. 173{182 (2015). DOI 10.1145/2756406.2756918.
URL http://doi.acm.org/10.1145/2756406.2756918

[17] NIST Digital Library of Mathematical Functions. https://dlmf.nist.gov/. Accessed June 12, 2018

[18] B.R. Miller, A. Youssef, "Technical aspects of the digital library of mathematical functions", Annals of Mathematics and Articial Intelligence 38(1), 121 (2003). DOI 10.1023/A:1022967814992.
URL https://doi.org/10.1023/A:1022967814992

[19] J. Misutka, L. Galambos, "Extending full text search engine for mathematical content", Towards Digital Mathematics Library. pp. 55-67 (2008)

[20] R. Miner, R. Munavalli," An approach to mathematical search through query formulation and data normalization", Towards Mechanized Mathematical Assistants pp. 342-355 (2007)

[21] P. Sojka, M. Liska, "Indexing and Searching Mathematics in Digital Libraries",Intelligent Computer Mathematics, ed. by J.H. Davenport, W.M. Farmer, J. Ur-ban, F. Rabe (Springer Berlin Heidelberg, Berlin, Heidelberg, 2011), pp. 228-243

[22] M. Kohlhase, I. Sucan," A search engine for mathematical formulae, Artificial Intelligence and Symbolic Computation" pp. 241-253 (2006)

[23] P. Graf, "Substitution tree indexing", Rewriting Techniques and Applications pp. 117-131(1995)

[24] R.Z. Thomas Schellenberg, Bo Yuan," Layout-based substitution tree indexing and retrieval for mathematical expressions", Proc.SPIE 8297, 8297 (2012). DOI 10.1117/12912502.
URL https://doi.org/10.1117/12.912502

[25] Zhang Q., Youssef A. ," Performance Evaluation and Optimization of Math-Similarity Search". In: Kerber M., Carette J., Kaliszyk C., Rabe F., Sorge V. (eds) Intelligent Computer Mathematics. CICM 2015. Lecture Notes in Computer Science, vol 9150. Springer, Cham (2015)

[26] P. Sojka, M. Liska, "The art of mathematics retrieval", Proceedings of the 11th ACM Symposium on Document Engineering pp. 57-60 (2011). DOI 10.1145/2034691.2034703.
URL http://doi.acm.org/10.1145/2034691.2034703

[27] X. Hu, L. Gao, X. Lin, Z. Tang, X. Lin, J.B. Baker, "WikiMirs: A Mathematical Information Retrieval System for Wikipedia", pp. 11-20 (2013).
DOI 10.1145/2467696.2467699.
URL http://doi.acm.org/10.1145/2467696.2467699.

[28] WikiMirs3. Koala. http://www.icst.pku.edu.cn/cpdp/WikiMirs3/. Query sets were downloaded, Accessed March 1, 2018.

[29] Mirjalili, S., Mirjalili, S. M., & Lewis, "A.. Grey wolf optimizer". Advances in Engineering Software, 69, 46–61. (2014)

[30] Wen Long, Jianjun Jiao, Ximing Liang, Mingzhu Tang, "Inspired grey wolf optimizer for solving large-scale function optimization problems Applied Mathematical Modelling", Volume 60, (August 2018), Pages 112-126.

[31] Mehak Kohli, Sankalap Arora, "Chaotic grey wolf optimization algorithm for constrained optimization problems" Journal of Computational Design and Engineering, In press, corrected proof, Available online (7 March 2017).

[32] Dhar Sourish, Roy Sudipta,"A Pattern Based Trie Indexing Scheme for Scientific Document Retrieval", (Unpublished and Communicated, 2018)

[33] Apache Lucene Core. https://lucene.apache.org/core/. Accessed June 12, 2018

[34]WikiMirs3.Koala.
http://www.icst.pku.edu.cn/cpdp/WikiMirs3/.
Querysets were downloaded, Accessed March 1, 2018

[35] J. Datta, A mtech seminar report : Ranking in information retrieval. Tech. rep., Department of Computer Science and Engineering, Indian Institute of Technology, Bombay (2013)

 DOI : 10.18231/2454-9150.2018.0639