Journal of King Saud University – Computer and Information Sciences 34 (2022) 6057–6065

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Multi-document extractive text summarization based on firefly algorithm

Minakshi Tomer [a,b,*], Manoj Kumar [c]

[a] University School of Information & Communication Technology, Guru Gobind Singh Indraprastha University, Delhi, India
[b] Information Technology Department, Maharaja Surajmal Institute of Technology, Delhi, India
[c] Netaji Subhas University of Technology, East Campus (Formerly Ambedkar Institute of Advanced Communication Technologies and Research) Delhi, India

ARTICLE INFO

Article history:
Received 22 January 2021
Revised 23 March 2021
Accepted 12 April 2021
Available online 24 April 2021

Keywords:
Multi-document text summarization
Meta-heuristic approaches
Extractive text summarization
Fitness function
DUC datasets and ROUGE

ABSTRACT

Extracting relevant information from a large amount of data is a challenging task. Automatic text summarization is a potential solution for obtaining this information. In this paper, a nature inspired swarm intelligence-based algorithm viz. firefly algorithm for multi-document text summarization is proposed. A new fitness function consisting of three features viz. topic relation factor, cohesion factor and readability factor is utilized. The experiments are performed on datasets from Document Understanding Conference i.e. DUC-2002, DUC-2003 and DUC-2004. The performance of the algorithm has been evaluated using ROUGE score. The performance of the proposed algorithm is compared with some other nature inspired ones such as particle swarm optimization (PSO) and genetic algorithm (GA). The performance of the proposed algorithm outperforms the other adopted ones.

1. Introduction

The information on world wide web is growing with an exponential rate (Amini et al., 2005; Khan et al., 2015). Therefore, it is necessary to provide the succinct form of the required information without losing its significance. Thus, reducing the reading time is a desirable prospect as it can greatly decrease human effort as well as help in finding the most important parts of any document or corpus of documents (Lloret and Palomar, 2012). Automatic text summarization (ATS) has been a successful solution for generating the shorter version of the input document without losing its main content. ATS can be categorized on the basis of its output as extractive and abstractive text summarization (Lloret and Palomar, 2012; Gambhir and Gupta, 2017). An extractive approach generates the summary by extracting the most significant sentences from the document, whereas abstractive approach generates the summary formed by some new words, new phrases, or new sentences (Gambhir and Gupta, 2017). ATS can also be classified based on input as single document summarization and multiple documents summarization (Lloret and Palomar, 2012; Gambhir and Gupta, 2017). Multi-document summarization has more challenges as compared to single document summarization (Goldstein et al., 2000). It includes issues like multiple documents with redundant information, compression of multiple document and speed of sentence selection with its extraction (Verma and Om, 2019). These issues are resolved using statistical tools and optimization techniques (Rautray and Balabantaray, 2017). While summarizing the document, keeping the relevancy and redundancy under control is an important task for any automatic summarizer (Verma and Om, 2019).

Many bio-inspired algorithms such as Fruit Fly Optimization (FOA) (Peng et al., 2020), Advanced Backtracking Search Algorithm (ABSO) (Wang et al., 2020), and Differential Evolution algorithm (DE) (Civicioglu and Besdok, 2021) are used for solving complex problems. Few of these meta-heuristic optimization approaches like Genetic Algorithm (GA) (Gordon, 1988; Kogilavani and Balasubramanie, 2010), Harmony Search Algorithm (HSA) (Shareghi and Hassanabadi, 2008), Particle Swarm Optimization (PSO) (Alguliev et al., 2011; Asgari et al., 2014), Cat Swarm Optimization (CSO) (Rautray and Balabantaray, 2017), Cuckoo Search (Rautray and Balabantaray, 2018), Firefly Algorithm (FA) and Shark Smell Optimization (SSO) (Abedinia et al., 2016) have achieved

* Corresponding author at: University School of Information & Communication Technology, Guru Gobind Singh Indraprastha University, Delhi, India.

E-mail addresses: tomer.minakshi@gmail.com (M. Tomer), manoj.kumar@nsut.ac.in (M. Kumar).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2021.04.004
1319-1578/

success in the field of text summarization for single document or multi-document summarization (Verma and Om, 2019). These existing methods mainly focused on maximum coverage with minimal redundancy. Firefly algorithm is stated to be a very efficient algorithm (Gandomi et al., 2011; Yang, 2009; Yang, 2010; Yang, 2013). There are many recently published review papers on firefly algorithm (Yang, 2014; Fister et al., 2014; Ali et al., 2014; Abdelaziz et al., 2015; Ariyaratne et al., 2015). Few modifications were performed on firefly algorithm to increase its performance (Tilahun et al., 2019). There are many research papers which indicate how successfully firefly has been applied in different areas like engineering application, medical application and economics application etc (Pan et al., 2013; Sekhar, 2014; Tilahun and Ong, 2013; Alweshah, 2014; Kwiecień and Filipowicz, 2012; Poursalehi et al., 2013).

In this paper, an automated text summarization algorithm for multi-document using optimization technique is proposed. In this proposed algorithm a new extractive text summarization is addressed based on firefly algorithm utilizing Topic Relation Factor (TRF), Cohesion Factor (CF) and Readability Factor (RF) as fitness function (Qazvinian et al., 2008). This algorithm is different from other methods as it utilizes new fitness function to evaluate features as compared to simple cosine function. This is performed to improve the results over the simple cosine function. These factors help in generating a summary in which the sentences are highly co-related to the topic (Silla et al., 2004; Salton and Buckley, 1988) and each sentence relates the same information (Mitra et al., 1997) while maintaining the flow of the summary.

The databases used for testing this algorithm are the (Document Understanding Conferences) DUC-2002, DUC-2003 and DUC-2004. The metrics used to measure the performance are (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004) Rouge-1, Rouge-2, Rouge-L and Rouge-SU4 on DUC-2002 and Rouge-1, Rouge-2 on DUC-2003, DUC-2004. Other metrics like recall, precision and f-measure are also computed for the proposed algorithm. The system summaries produced by the proposed algorithm are compared against many human generated reference summaries and their average values are also being calculated.

The rest of the paper is organized as follows: The literature survey of text summarization using meta-heuristic approaches is presentd in Section 2. Section 3 explains the proposed algorithm in detail followed by the results of the experiments which are presented in Section 4. Finally, Section 5 presents the conclusions and future works.

2. Literature survey

In literature, many techniques have been successfully applied in the area of text summarization including statistical-approaches, discourse-approaches, topic-approaches, graph-based, machine learning (Lloret and Palomar, 2012) and meta-heuristic approaches (Verma and Om, 2019). Another summarization techniques are ensembled approaches that combines extractive and abstractive approaches (Tomer and Kumar, 2020; Sharma et al., 2020). This section presents the survey of multi-document summarization using meta-heuristic approaches.

Genetic algorithm was the first meta-heuristic algorithm that has been applied in multi-document text summarization for the retrieval of most significant sentences (He et al., 2006). A sentence selection approach was proposed in Qazvinian et al. (2008) where genetic algorithm was utilized. Genetic algorithm was used to create a summary which was later evaluated using fitness function based on topic relation factor, readability factor and cohesion factor. Documents were represented using a DAG (Directed Acyclic Graph) in which each sentence is represented as a vertex. The weight of edges shows the similarity of the sentences and the TF-IDF (Term frequency-Inverse document frequency) weighing system was used for representing the sentences.

A novel text clustering technique called ensemble clustering method for text summarization was proposed in Lee et al. (2017). Both genetic algorithms and particle swarm optimization were combined to get the results. Similarity of sentences were calculated using normalized google distance. Automatic population partitioning utilized GA + PSO (genetic algorithm + particle swarm optimization) for sentence clustering. PSO is applied to good global optimum candidates with high fitness value. GA is applied on individuals having less likelihood of achieving good fitness evaluation. Top ranked sentences were selected from each cluster and were then sorted.

A cat swarm optimization algorithm for multi-document summarization was proposed in Rautray and Balabantaray (2017). The model was compared with harmonic search algorithm-based summarizer and particle swarm-based summarizer. A sentence informative score was calculated to represent the weight of a sentence whereas cosine similarity was utilized to generate the inter-sentence similarity score. Another algorithm (Rautray and Balabantaray, 2018) presents cuckoo search algorithm for text summarization on multiple documents. The cuckoo search-based summarizer was compared with PSO and ACO (Ant colony optimization) for multi-document on DUC dataset.

An ATS (Automatic Text Summarization) model for single and multi-document based on fuzzy with firefly algorithm (FA) was proposed in Ali and Malallah (2017). Two main factors were considered while summarizing the text: relevance and redundancy. The model consists of four steps: first is pre-processing in which tokenization and stemming is done, followed by feature extraction in second step which calculate the scores for each sentence. Fuzzy logic is applied in the third step by labeling of score as high, medium and low. In final step, association-rule mining is performed using the firefly algorithm. The database used was TAC-2011 and rouge was used as evaluation metric.

Firefly algorithm (FA) was also used for arabic text summarization (Al-Abdallah and Al-Taani, 2019). The proposed approach achieved better rouge score in comparison to genetic algorithm and harmony search algorithm.

A survey of various text summarization approaches, especially focusing on swarm intelligence with special focus on Ant Colony Optimization (ACO) in Mosa et al. (2019) was performed. Ant colony optimization was recommended as it has higher accuracy and better convergence and stability. Single document, multi-document and short-text summarization were studied. Jaccard similarity and cosine similarity were used for the fitness function.

A shark smell optimization method based on multi-document for summarization was proposed in Verma and Om (2019). The main features focused were coverage, non-redundancy and relevancy to generate a better summary. A linear combination of word embedding and google distance based similarity methods were utilized for identifying features. Experiments were performed on six benchmark datasets and compared with twelve others methods.

3. Proposed framework

In the literature survey it has been observed that meta-heuristic approaches are frequently utilized to enhance the performance of multi-document text summarization. Inspired by these methods, an algorithm for multi-document text summarization that uses topic relation factor, cohesion factor and readability factor as fitness function is proposed in this section. This proposed algorithm combines features (attributes observable in a text document) that help in generation of a summary that is related to the topic, has a flow, has high co-relation between sentences and is highly readable. It is composed of the following steps: (i) Pre-Processing, (ii) Document Representation (iii) Summary Scoring/Fitness Function and (iv) Utilization of Firefly Algorithm. The flowchart for proposed automatic text summarization is displayed in Fig. 1.

3.1. Pre-processing

It is a process in which the documents are converted into a form suitable for performing text summarization efficiently. This includes case conversion, tokenization, stop word removal and stemming.

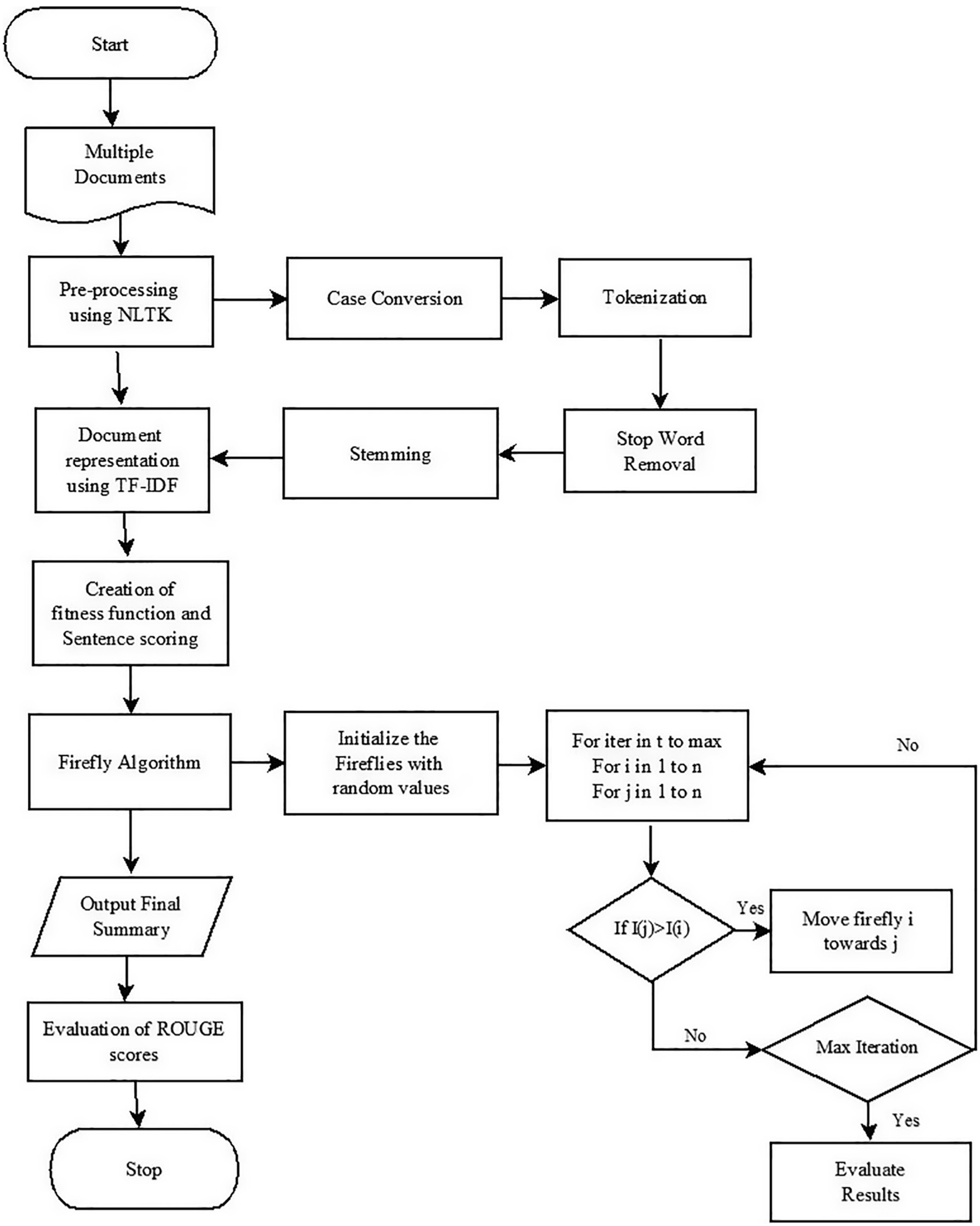

Fig. 1. Proposed ATS Framework.

3.1.1. Case conversion

All the characters in the document D are changed to either upper-case or lower-case for uniformity purposes.

3.1.2. Tokenization

Breaking each sentence into a continuous stream of tokens as T= $t_1, t_2, t_3.......t_n$, where tokens $t_1, t_2, \ldots t_n$ represent individual tokens in the document D. It also identifies the start and end of a sentence. Tokenization prepares the document for the following two steps i.e. stop word removal and stemming by making it easier to analyze and perform operations on the text.

3.1.3. Stop word removal

In this process, stop words can be removed as they don't have any effect in extracting the significant tokens or the important weighted words.

3.1.4. Stemming

It is a process in which words that originate from the same root or base or stem word are identified and replaced by that word. For example, the words "write", "writing", "written", "wrote", etc. are transformed into their root word which is "write" by using the Word Net lemmatizer.

3.2. Document representation

TF-IDF stands for Term Frequency-Inverse Document Frequency which is a common method to transform text into a meaningful representation of numbers. The Term Frequency (TF) is the number of times each term i occurs in each document j. It is given by the formula (1):

$$TF_{i,j} = freq_{i,j}/maxfreq \tag{1}$$

where, $TF_{i,j}$is the TF(Term Frequency) score,

$freq_{i,j}$ is the frequency of the i^{th} term in j^{th} document,

$maxfreq$ is the highest frequency of any term among all document.

Inverse Document Frequency (IDF) represents the relation between the total number of sentences N and the number of sentences in which a term occurs n_i. It is given by the formula (2):

$$IDF_i = log(N/n_i) \tag{2}$$

Finally, the TF-IDF value is given by the formula mention in Eq. (3)

$$W_{i,j} = TF_{i,j} * IDF_i \tag{3}$$

The TF-IDF value or weighting value of all the terms are stored in a matrix for further use during the text summarization process.

3.3. Summary scoring/Fitness function

A good summary is a readable summary that covers all the topics covered in the actual document. The sentences in the summary should be highly related to each other and there should be a flow to the summary. To achieve these three features have been used – (i) Topic Relation Factor (TRF), (ii) Cohesion Factor (CF) (iii) Readability Factor (RF).

3.3.1. Topic Relation Factor (TRF)

Topic Relation Factor (TRF) explains how well the summary is related to the topic. To calculate the Topic Relation Factor, the average similarity of sentences in the summary divided by the maximum average can be considered. TR is the average similarity to title in the summary as in Eq. (4):

$$TR = \frac{\sum_{s_j \epsilon summary} sim(s_j, q)}{S} \tag{4}$$

Here, s_j is the j^{th} sentence in the summary, q is the title and S is the total number of sentences in the summary. This value can be normalized by dividing it by the maximum TR value calculated for all the generated summaries.Thus, the topic relation factor is given by the following Eq. (5):

$$TRF = \frac{TR}{max_{\forall summary} TR} \tag{5}$$

3.3.2. Cohesion Factor (CF)

Cohesion Factor (CF) determines whether sentences in the summary are talking about the same topic or not. To calculate the similarity between every pair of sentences, the set of sentences is represented as a graph. It also determines whether the summary makes sense as only the $CF_{i,j}$ value is calculated in which the sentence j occurs chronologically after the sentence i in the original document. Thus there are two assumptions as in Eq. (6) and (7).

$$\forall_i < N : sim(s_i, s_i) = 0 \tag{6}$$

$$\forall_{i,j} < N : sim(s_j, s_i) = 0 \ if \ s_i \ occurs \ before \ s_j \ in \ the \ document \tag{7}$$

N is the total number of sentences in the document. To calculate CF, two terms C and M need to be defined first. C is the average of similarities of all sentences in the summary as in Eq. (8).

$$C = \frac{\sum_{\forall s_i s_j \epsilon summarysubgraph} W(s_i s_j)}{N_s} \tag{8}$$

where, $W(s_i, s_j)$ is the weight of the path from sentence i to sentence j and N_s is the total number of edges in the summary sub-graph. The summary graph is considered to represent the Cohesion Factor (CF) of the fitness function.

N_s can be calculated as in Eq. (9)

$$N_s = (S-1) + (S-2) + \ldots\ldots.. = \frac{(S)*(S-1)}{2} \tag{9}$$

M is the maximum weight in the graph i.e. the maximum similarity of the sentences as calculated in Eq. (10).

$$M = maxSim_{i,j} \qquad i,j \leqslant N \tag{10}$$

Finally, CF is given as Qazvinian et al. (2008) in Eq. (11). The logarithm will help to avoid the low magnitude of CF when average is very smaller than maximum.

$$CF = \frac{log(C*9+1)}{log(M*9+1)} \tag{11}$$

The base of the logarithm is 10.

3.3.3. Readability Factor (RF)

A readable summary is a one, in which sentences have high similarity with its next sentence. For example, first sentence should have high similarity with second sentence, same for sentence two and three and so on. This ensures that a smooth summary is generated. The readability of a summary with length S is given by (12)

$$R = \sum W(s_i, s_{i+1}) \tag{12}$$

And the readability factor (RF) is given as (13)

$$RF = \frac{R}{max_{\forall} R_i} \tag{13}$$

The maximum value of R_i is the longest path with a definite number of nodes in the weight matrix. To find the smoothest summary of length k just look up the maximum value in the k^{th} column of the weight matrix.

3.3.4. Fitness function

The fitness function is constructed using the three factors explained above. It has constants whose values can be changed to fit the user's need. This means that the fitness function can be adjusted to give a more readable summary or a summary with higher similarity to the title, etc as presented in Eq. (14).

$$F = \frac{(\alpha * TRF) + (\beta * CF) + (\gamma * RF)}{\alpha + \beta + \gamma} \tag{14}$$

α, β and γ are real numbers that are defined by the user. Since the values of TRF, CF and RF vary between 0 and 1, the final value of the fitness function also varies between 0 and 1.

3.4. Proposed algorithm for text summarization

The proposed FbTS (firefly based text summarization) utilize the firefly algorithm to do the text summarization. Firefly algorithm is a swarm-based algorithm which is easy to understand and use. It was proposed by Yang (2008) and is based on the mating behavior of fireflies. Firefly algorithm is based on the concept of firefly being attracted to brightest/brighter firefly. The brightest firefly represents the optimal solution whereas remaining fireflies are non-optimal solution. The fireflies move towards brighter fireflies, similarly non-optimal solutions are moved towards optimal solution.

3.4.1. Representation of firefly in proposed algorithm

Each firefly is represented by vector of size N, where N is the total number of sentences present in all the documents combined. If a sentence is to be included in the summary, it is labeled as 1 otherwise it is labeled as 0. There are K such fireflies initialized randomly, the light intensity is calculated for each firefly using the fitness function. In the proposed algorithm, three features TRF, CF and RF are used as a fitness function which calculates the fitness value of all the fireflies. The maximum value of the fitness function is considered the brightest firefly and this firefly is the one that other fireflies will be attracted towards.

3.4.2. Use of firefly algorithm

The position of each firefly is updated based on the light intensity of other fireflies in the population using the Eq. (15).

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2}(x_j - x_i) + \alpha(\epsilon() - 0.5) \tag{15}$$

β_0 is the absolute brightness of the firefly i.e. brightness at r = 0. This value is usually set to 1 (Tilahun et al., 2019). γ is used to adjust the distance between two fireflies. α is the control parameter for the random movement of firefly. $\epsilon()$ is a random vector with values between 0 and 1. For the brightest firefly x_j, the second term in Eq. (15) is neglected as shown in Eq. (16).

$$x_j = x_j + \alpha(\epsilon() - 0.5) \tag{16}$$

3.4.3. Calculations on firefly

For summary representation purpose, any value greater than 0.5 will be considered as representing 1 in the vector i.e. the sentence at that position will be considered as part of the summary and any value less than 0.5 will be considered as 0 and the sentence at that position will be excluded from the summary. This conversion is only done in calculation of the intermediate summaries.

3.4.4. Summary generation

Finally, after the maximum iterations, the final summary has to be generated. The best firefly i.e. the one with the best fitness value is selected and then the final summary is generated. For this, the values in the best firefly are sorted and then the sentence position containing the best values are selected. This is done till the maximum limit of sentences or words is not reached. After that, the final summary is obtained of all the documents given at the time of input. The proposed FbTS algorithm with time complexity $O(n^2t)$ for text summarization is explained in Fig. 2.

4. Experiments and results

This section includes the implementation of the adopted algorithm on various datasets to check the performance of the FbTS algorithm and its comparison to other methods. The FbTS algorithm is compared with genetic algorithm, particle swarm optimization, ant colony optimization and many other methods using DUC-2002, DUC-2003 and DUC-2004 datasets. All the experiments are done on PyCharm (Version- 2018.1.4) on a system with 64-bit Windows 10 operating system having intel i3@ 2.3 GHz dual core processor (4 GB RAM). The python version was Python 3.6. The analysis of the results after generating the final summary was done using ROUGE toolkit (Lin and Hovy, 2003) in terms of Rouge-1, Rouge-2, Rouge-L and Rouge-SU4 score.

4.1. Dataset

The dataset used for testing the automatic text summarizer is the DUC-2002, DUC-2003 and DUC-2004. It is a database which was issued by the DUC (Document Understanding Conference). It was generated by the National Institute of Standards and Technology (NIST) to encourage researchers for conducting large-scale experiments and facilitate progress in the field of summarization. The URL of the dataset is*http://www-nlpir.nist.gov/projects/duc/data*. Table 1 provide the description of datasets.

4.2. Controlling parameters

The FbTS algorithm requires setting of the parameters which includes number of fireflies and number of iterations. As mentioned (Verma and Om, 2019) there is no definite rule to set the parameters. The number of iterations provides the best intensity within one generation. Therefore, after the trial procedures the number of iterations was chosen to be 100. As most of the studies suggested (Yang, 2013) the attractive coefficient (β_0) and absorption coefficient (γ_0) is set to be 1. The controlling parameters used in the proposed algorithm are presented in Table 2.

4.3. Evaluation of the proposed algorithm

The proposed algorithm was implemented and the results were evaluated on the DUC-2002, DUC-2003 and DUC-2004 datasets. The metric used were the Rouge-1, Rouge-2, Rouge-L and the Rouge-SU4 for DUC-2002. The average recall, precision and the F-score were calculated for each ROUGE score as mentioned in Table 3. Whereas, for DUC-2003, DUC-2004 datasets Rouge-1 and Rouge-2 scores were calculated.

4.4. Comparison between the proposed algorithm and the other adopted ones

On DUC 2002: In this paper Firefly Algorithm with JS-Divergence (Yang, 2009), Genetic Algorithm with JS-divergence and cosine similarity (Ali and Malallah, 2017), Particle Swarm

1. *Objective function is represented as f(x) which is represented by a function using three features – TRF, CF and RF*
2. *Initial population of fireflies is generated x_i (i= 1,2, ..., 50), and each firefly is represented by a binary vector where 1 represents the presence of that sentence in the summary and 0 represents its absence.*
3. *Light intensity I_i at x_i is determined by $f(x_i)$*
4. *Define light absorption co-efficient(γ)*
5. ***while** (t < Maximum NumberOfIterations (100))*
6. ***for i** = 1: n all n fireflies*
7.   ***for j** = 1: i all n fireflies*
8.   ***if** ($Intensity_j$> $Intensity_i$), Firefly i is moved towards j in d-dimension;*

           endif
9.   *Attractiveness varies with distance r via $exp^{[-\gamma r]}$*
10.   *New binary vectors are evaluated for each firefly and light intensity is updated*
11.   ***end for j***
12. ***end for i***
13. *The fireflies are ranked and the current best firefly is found*
14. *The current best firefly is updated*
15. ***end while***
16. *Generation of final summary is done by selecting the top fireflies and evaluation of result is done using ROUGE score*

Fig. 2. FbTS algorithm for automatic text summarization.

Table 1
Datasets description.

Dataset description	DUC-2002	DUC-2003	DUC-2004
Sets of Documents	60	30	100
Documents per set	10	10	10
Data source	duc.nist.gov	duc.nist.gov	TREC
Summary Length(words)	100	100	665 bytes

Table 2
Parameters used for proposed FbTS algorithm and other methods

Parameters for proposed Algorithm	Vaule
The population size	50
Number of Iterations	100
Attractiveness Coefficient β_0	1
Absorption Coefficient γ_0	1
Maximum Radius of the Random Step α_0	3
Random Step ϵ_0	[0,1]

Table 3
Recall, Precision and F-score for the FbTS algorithm on DUC-2002

Rouge Type	Average Recall	Average Precision	Average F-score
Rouge-1	0.43803	0.48095	0.47821
Rouge-2	0.21212	0.25012	0.22951
Rouge-L	0.33333	0.33929	0.33628
Rouge-SU4	0.19344	0.23506	0.21223

Optimization with JS-divergence and cosine similarity (Asgari et al., 2014) and Ant colony Optimization with cosine similarity have been implemented on DUC-2002 dataset. FbTS algorithm is compared with other methods using Rouge scores. The TF-IDF, LexRank, KL-Greedy, JS-Greedy and ICSI implemented in Peyrard and Eckle-Kohler (2016) are also included in Table 4 to be compared against our proposed algorithm. The results show that the performance of the proposed algorithm has better Rouge-1 (0.47821) and Rouge-2 (0.22951) score in comparison to other methods as shown in Fig. 3.

On DUC-2003: The proposed algorithm is implemented and compared with Genetic algorithm with JS-divergence, PSO with JS-divergence, ACO with cosine similarity, TF-IDF, LexRank, KL-Greedy, JS-Greedy and ICSI (Peyrard and Eckle-Kohler, 2016) on DUC-2003 dataset. The proposed algorithm shows better Rouge-1 (0.4419) and Rouge-2 (0.1602) scores on the dataset. The results are mentioned in Table 5 and Fig. 4.

Table 4
Performance comparison of FbTS algorithm with other methods on DUC-2002

		EvaluationMetric	
Methods	Fitness Function	Rouge-1	Rouge-2
FA (JS -Divergence)	JS-Divergence	0.3262	0.1527
GA (JS-Divergence)	JS-Divergence	0.4117	0.1758
GA (Cosine Similarity)	CosineSimilarity	0.3597	0.1374
PSO (JS-Divergence)	JS-Divergence	0.3759	0.1312
PSO (Cosine Similarity)	CosineSimilarity	0.3235	0.1835
ACO (Cosine Similarity)	CosineSimilarity	0.3289	0.1589
TF-IDF	NA	0.4072	0.1201
Lex Rank	NA	0.4311	0.1388
KL-Greedy	NA	0.3945	0.1125
JS-Greedy	NA	0.4299	0.1455
ICSI	NA	0.4434	0.1556
FbTS(TRF, CF, RF)	**TRF,CF and RF**	**0.4782**	**0.2295**

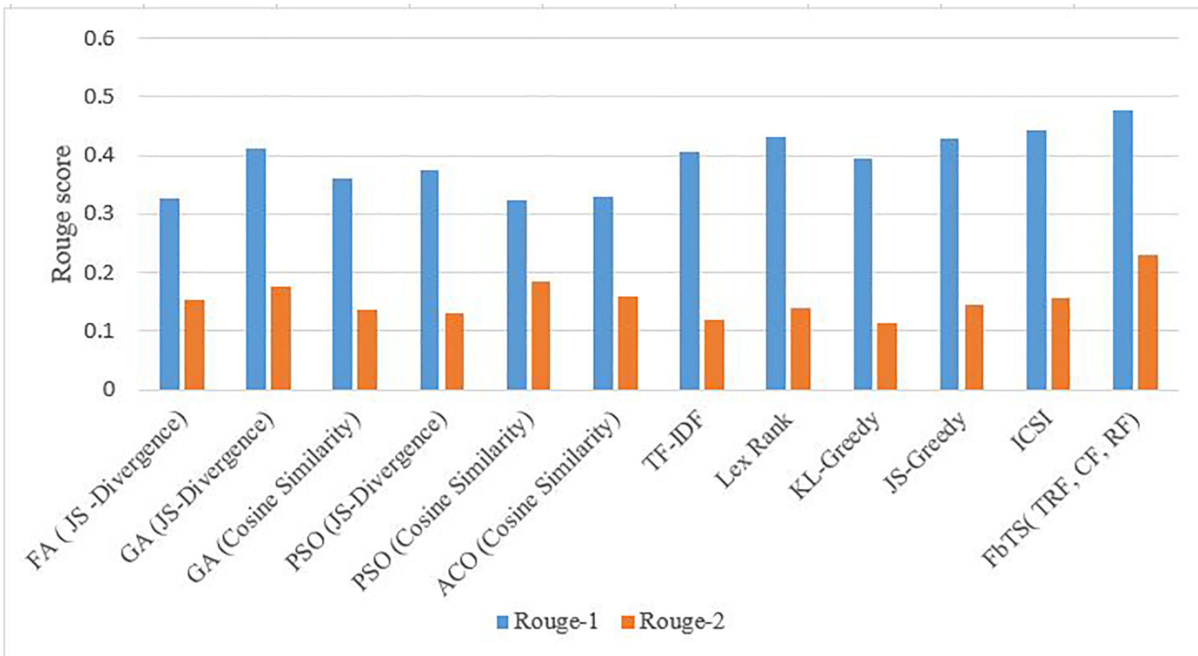

Fig. 3. Rouge score comparison of FbTS algorithm with other methods on DUC-2002 dataset.

Table 5
Performance comparison of FbTS Algorithm with other methods on DUC-2003

		Evaluation Metric	
Methods	Fitness Function	Rouge-1	Rouge-2
GA(JS-Divergence)	JS-Divergence	0.4397	0.1416
PSO(JS-Divergence)	JS-Divergence	0.4321	0.1507
ACO (CosineSimilarity)	Cosine Similarity	0.4231	0.1407
TF-IDF	NA	0.3222	0.066
LexRank	NA	0.3574	0.0793
KL-Greedy	NA	0.3231	0.0715
JS-Greedy	NA	0.3312	0.064
ICSI	NA	0.3763	0.0947
FbTS (TRF, CF, RF)	**TRF,CF and RF**	**0.4419**	**0.1602**

Table 6
Performance comparison of FbTS Algorithm with other methods on DUC-2004

		Evaluation Metric	
Methods	Fitness Function	Rouge-1	Rouge-2
GA (JS-Divergence)	JS-Divergence	0.3546	0.0937
PSO (JS-divergence)	JS-Divergence	0.3521	0.0935
ACO (Cosine Similarity)	CosineSimilarity	0.3542	0.0837
MCRMR_PSO (Cosine Similarity)	CosineSimilarity	0.385	0.139
MCRMR_SSO (Cosine Similarity)	CosineSimilarity	0.410	0.136
MMR_SE	NA	0.336	0.099
System 61(Centroid score)	Centroidscore	0.359	0.091
LexRank	NA	0.326	0.079
FbTS (TRF, CF, RF)	**TRF,CF and RF**	**0.4244**	**0.1764**

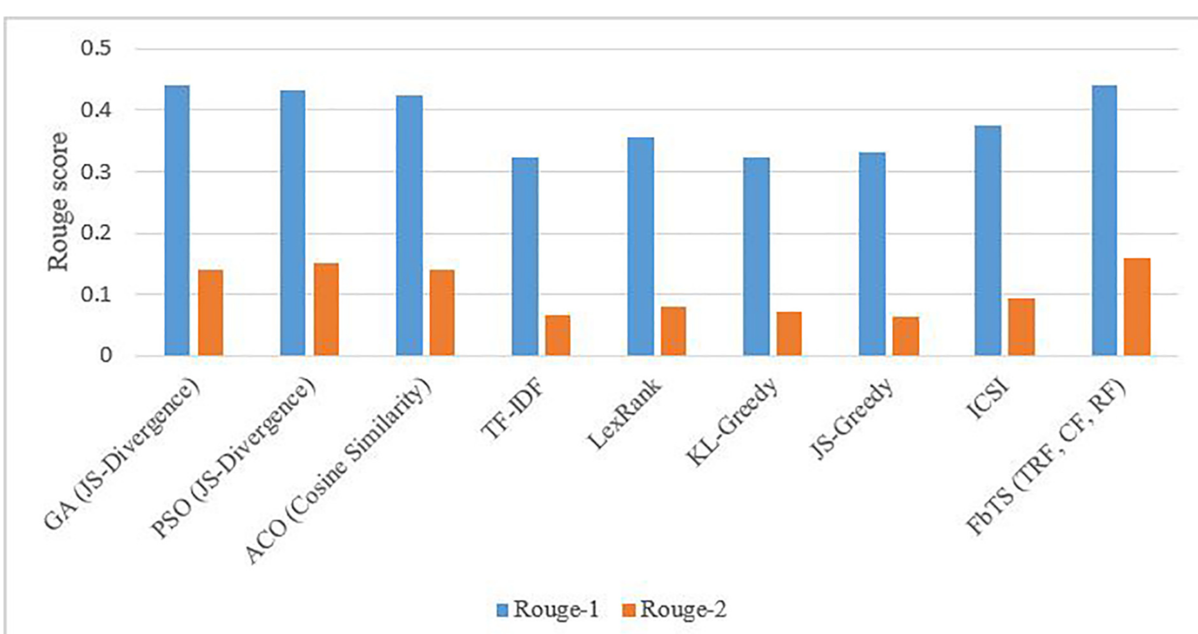

Fig. 4. Rouge score comparison of FbTS algorithm with other methods on DUC-2003 dataset.

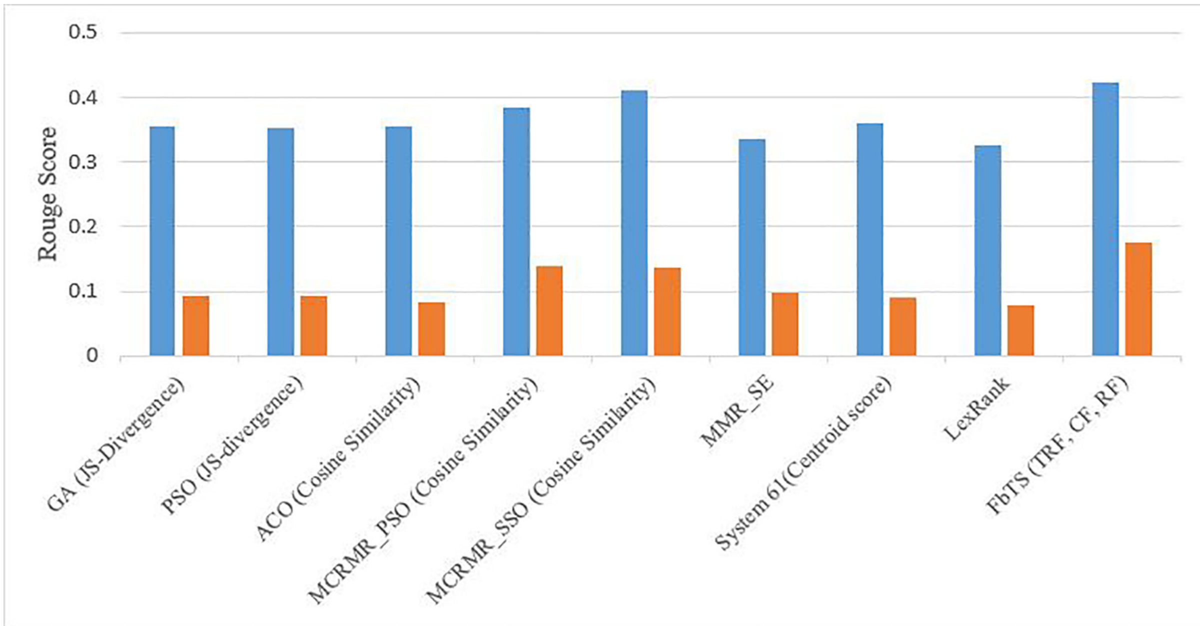

Fig. 5. Rouge score comparison of FbTS algorithm with other methods on DUC-2004 dataset.

On DUC-2004: The proposed algorithm is implemented and compared with Genetic algorithm with JS-divergence, PSO with JS-divergence, ACO with cosine similarity, $MCRMR - PSO$ with cosine similarity, $MCRMR - SSO$ with cosine similarity, $MMR - SE$ (Verma and Om, 2019), System 61 with centroid score and LexRank on DUC-2004 dataset. The proposed algorithm shows better result using Rouge-1 (0.4244) and Rouge-2 (0.1764) scores on DUC-2004 dataset as in Table 6 and Fig. 5.

5. Conclusion

In this paper, a novel FbTS metaheuristic-based algorithm for multi-document text summarization is proposed. A new fitness function based on Topic Relation Factor (TRF), Cohesion Factor (CF) and Readability Factor (RF) are utilized to calculate the score of each sentence. The sentences with best score are selected to generate the summary. However, the selection of TRF, CF and RF as a fitness function improved the quality of resultant extractive summary. To confirm the performance of the FbTS algorithm, several experimentations were performed on standard DUC-2002, DUC-2003 and DUC-2004 datasets. The experimental result was evaluated using ROUGE score. The proposed FbTS algorithm showed a higher ROUGE-1 and ROUGE-2 score than the other nature inspired ones, genetic algorithm and particle swarm optimization.

In future, new feature selection methods, new fitness functions can be used with bio-inspired algorithm to improve the quality of extractive summary. These extractive methods can be merged with deep neural based model for abstractive summary. These hybrid models (extractive and abstractive) may give better results in terms of performance.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abdelaziz, A., Mekhamer, S., Badr, M., Algabalawy, M., 2015. The firefly metaheuristic algorithms: Developments and applications. International Electrical Engineering Journal (IEEJ) 6 (7), 1945–1952.
Abedinia, O., Amjady, N., Ghasemi, A., 2016. A new metaheuristic algorithm based on shark smell optimization. Complexity 21 (5), 97–116.
Al-Abdallah, R.Z., Al-Taani, A.T., 2019. Arabic text summarization using firefly algorithm. In: 2019 Amity International Conference on Artificial Intelligence (AICAI), IEEE, pp. 61–65.
Alguliev, R.M., Aliguliyev, R.M., Hajirahimova, M.S., Mehdiyev, C.A., 2011. Mcmr: Maximum coverage and minimum redundant text summarization model. Expert Systems with Applications 38 (12), 14514–14522.
Ali, Z.H., Malallah, P., 2017. A, Multi-document summarization using fuzzy logic and firefly algorithm. Journal of College of Education 1 (3), 139–152.
Ali, N., Othman, M.A., Husain, M.N., Misran, M.H., 2014. A review of firefly algorithm. ARPN Journal of Engineering and Applied Sciences 9 (10), 1732–1736.
Alweshah, M., 2014. Firefly algorithm with artificial neural network for time series problems. Research Journal of Applied Sciences, Engineering and Technology 7 (19), 3978–3982.
Amini, M.R., Usunier, N., Gallinari, P., 2005. Automatic text summarization based on word-clusters and ranking algorithms. In: European Conference on Information Retrieval. Springer, pp. 142–156.
Ariyaratne, M., Pemarathne, W., 2015. A review of recent advancements of firefly algorithm; a modern nature inspired algorithm.
Asgari, H., Masoumi, B., Sheijani, O.S., 2014. Automatic text summarization based on multi-agent particle swarm optimization, in. In: 2014 Iranian Conference on Intelligent Systems (ICIS), IEEE, pp. 1–5.
Civicioglu, P., Besdok, E., 2021. Bezier search differential evolution algorithm for numerical function optimization: A comparative study with crmlsp, mvo, wa, shade and lshade. Expert Systems with Applications 165, 113875.
Fister, I., Yang, X.-S., Fister, D., 2014. Firefly algorithm: a brief review of the expanding literature. In: Cuckoo Search and Firefly Algorithm. Springer, pp. 347–360.
Gambhir, M., Gupta, V., 2017. Recent automatic text summarization techniques: a survey. Artificial Intelligence Review 47 (1), 1–66.
Gandomi, A.H., Yang, X.-S., Alavi, A.H., 2011. Mixed variable structural optimization using firefly algorithm. Computers & Structures 89 (23–24), 2325–2336.
Goldstein, J., Mittal, V.O., Carbonell, J.G., Kantrowitz, M., 2000. Multi-document summarization by sentence extraction, in. In: NAACL-ANLP 2000 Workshop: Automatic Summarization.
Gordon, M., 1988. Probabilistic and genetic algorithms in document retrieval. Communications of the ACM 31 (10), 1208–1218.
He, Y.-X., Liu, D.-X., Ji, D.-H., Yang, H., Teng, C., 2006. Msbga: A multi-document summarization system based on genetic algorithm. In: 2006 International Conference on Machine Learning and Cybernetics, IEEE, pp. 2659–2664.

Khan, A., Salim, N., Kumar, Y.J., 2015. A framework for multi-document abstractive summarization based on semantic role labelling. Applied Soft Computing 30, 737–747.

Kogilavani, A., Balasubramanie, P., 2010. Clustering based optimal summary generation using genetic algorithm, in. In: 2010 International Conference on Communication and Computational Intelligence (INCOCCI), IEEE, pp. 324–329.

Kwiecień, J., Filipowicz, B., 2012. Firefly algorithm in optimization of queueing systems. Bulletin of the Polish Academy of Sciences Technical Sciences 60 (2), 363–368.

Lee, J.S., Hah, H., Park, S.C., 2017. Less-redundant text summarization using ensemble clustering algorithm based on ga and pso, Wseas Transactions On Computers 16.

Lin, C.-Y., 2004. Rouge: A package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81.

Lin, C.-Y., Hovy, E., 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In: Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, pp. 150–157.

Lloret, E., Palomar, M., 2012. Text summarisation in progress: a literature review. Artificial Intelligence Review 37 (1), 1–41.

Mitra, M., Singhal, A., Buckley, C., 1997. Automatic text summarization by paragraph extraction. Intelligent Scalable Text Summarization.

Mosa, M.A., Anwar, A.S., Hamouda, A., 2019. A survey of multiple types of text summarization with their satellite contents based on swarm intelligence optimization algorithms. Knowledge-Based Systems 163, 518–532.

Pan, F., Ye, C., Wang, K., Cao, J., 2013. Research on the vehicle routing problem with time windows using firefly algorithm. JCP 8 (9), 2256–2261.

Peng, L., Zhu, Q., Lv, S.-X., Wang, L., 2020. Effective long short-term memory with fruit fly optimization algorithm for time series forecasting. Soft Computing, 1–21.

Peyrard, M., Eckle-Kohler, J., 2016. A general optimization framework for multi-document summarization using genetic algorithms and swarm intelligence. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 247–257.

Poursalehi, N., Zolfaghari, A., Minuchehr, A., Moghaddam, H., 2013. Continuous firefly algorithm applied to pwr core pattern enhancement. Nuclear Engineering and Design 258, 107–115.

Qazvinian, V., Hassanabadi, L.S., Halavati, R., 2008. Summarising text with a genetic algorithm-based sentence extraction. International Journal of Knowledge Management Studies 2 (4), 426–444.

Rautray, R., Balabantaray, R.C., 2017. Cat swarm optimization based evolutionary framework for multi document summarization. Physica A: Statistical Mechanics and its Applications 477, 174–186.

Rautray, R., Balabantaray, R.C., 2018. An evolutionary framework for multi document summarization using cuckoo search approach: Mdscsa. Applied Computing and Informatics 14 (2), 134–144.

Salton, G., Buckley, C., 1988. Term-weighting approaches in automatic text retrieval. Information Processing & Management 24 (5), 513–523.

Sekhar, J.C., 2014. Application of firefly algorithm for combined economic load and emission dispatch. International Journal on Recent and Innovation Trends in Computing and Communication 2 (8), 2448–2452.

Shareghi, E., Hassanabadi, L.S., 2008. Text summarization with harmony search algorithm-based sentence extraction, in. In: Proceedings of the 5th International Conference on Soft Computing as Transdisciplinary Science and Technology, pp. 226–231.

Sharma, B., Tomer, M., Kriti, K., 2020. Extractive text summarization using f-rbm. Journal of Statistics and Management Systems 23 (6), 1093–1104.

Silla, C.N., Pappa, G.L., Freitas, A.A., Kaestner, C.A., 2004. Automatic text summarization with genetic algorithm-based attribute selection. In: Ibero-American Conference on Artificial Intelligence. Springer, pp. 305–314.

Tilahun, S.L., Ong, H.C., 2013. Vector optimisation using fuzzy preference in evolutionary strategy based firefly algorithm. International Journal of Operational Research 16 (1), 81–95.

Tilahun, S.L., Ngnotchouye, J.M.T., Hamadneh, N.N., 2019. Continuous versions of firefly algorithm: a review. Artificial Intelligence Review 51 (3), 445–492.

Tomer, M., Kumar, M., 2020. Improving text summarization using ensembled approach based on fuzzy with lstm. Arabian Journal for Science and Engineering 45 (12), 10743–10754.

Verma, P., Om, H., 2019. Mcrmr: Maximum coverage and relevancy with minimal redundancy based multi-document summarization. Expert Systems with Applications 120, 43–56.

Wang, L., Peng, L., Wang, S., Liu, S., 2020. Advanced backtracking search optimization algorithm for a new joint replenishment problem under trade credit with grouping constraint. Applied Soft Computing 86, 105953.

Yang, X.-S. et al., 2008. Firefly algorithm. Nature-Inspired Metaheuristic Algorithms 20, 79–90.

Yang, X.-S., 2009. Firefly algorithms for multimodal optimization, in: International Symposium on Stochastic Algorithms, Springer, pp. 169–178.

Yang, X.-S., 2010. Firefly algorithm, levy flights and global optimization, in: Research and Development in Intelligent Systems XXVI, Springer, pp. 209–218.

Yang, X.-S., 2013. Multiobjective firefly algorithm for continuous optimization. Engineering with Computers 29 (2), 175–184.

Yang, X.-S., 2014. Nature-Inspired Optimization Algorithms. Elsevier.