Applied Computing and Informatics 14 (2018) 134–144

Contents lists available at ScienceDirect

Applied Computing and Informatics

journal homepage: www.sciencedirect.com

An evolutionary framework for multi document summarization using Cuckoo search approach: MDSCSA

Rasmita Rautray [a,*], Rakesh Chandra Balabantaray [b]

[a] Department of Computer Science and Engineering, Siksha 'O' Anusandhan University, Bhubaneswar 751030, Odisha, India
[b] Department of Computer Science, IIIT, Bhubaneswar, Odisha, India

A R T I C L E I N F O

Article history:
Received 21 February 2017
Revised 1 May 2017
Accepted 11 May 2017
Available online 13 May 2017

Keywords:
Text summarizer
Multi-document summarization
Extractive summary
Cuckoo search

A B S T R A C T

In today's scenario the rate of growth of information is expanding exponentially in the World Wide Web. As a result, extracting valid and useful information from a huge data has become a challenging issue. Recently text summarization is recognized as one of the solution to extract relevant information from large documents. Based on number of documents considered for summarization, the summarization task is categorized as single document or multi-document summarization. Rather than single document, multi-document summarization is more challenging for the researchers to find accurate summary from multiple documents. Hence in this study, a novel Cuckoo search based multi-document summarizer (MDSCSA) is proposed to address the problem of multi-document summarization. The proposed MDSCSA is also compared with two other nature inspired based summarization techniques such as Particle Swarm Optimization based summarization (PSOS) and Cat Swarm Optimization based summarization (CSOS). With respect to the benchmark dataset Document Understanding Conference (DUC) datasets, the performance of all algorithms are compared in terms of ROUGE score, inter sentence similarity and readability metric to validate non-redundancy, cohesiveness and readability of the summary respectively. The experimental analysis clearly reveals that the proposed approach outperforms the other summarizers included in this study.

1. Introduction

Now a day's the rate of information growth is expanding exponentially in the World Wide Web, which creates information overload problem. One solution to this problem is shortening of information, called text summarization (TS). Text summarization is the process of creating shorter version of original text without losing main contents [1] called summary. The summary provides a quick guide to create interest on information, helps in making decision on document whether it is readable or not as well as it is served as a time saver for users [2]. The way in which summary is generated either is an extraction or an abstraction method [3,4]. Extraction based summaries are generated by selecting the important portions of the original text. Whereas, abstraction based summaries requires linguistic analysis to construct new sentences from the original text [5,6]. Based on dimension, extraction based summaries can be categorized into two ways i.e., generic or query dependent [7]. Generic summary reflects the major content of the documents without any additional information. But, Query-dependent summary focuses on the information expressed in the given queries [8,9].

Number of documents considered for generating summary, can classify the summarization problem as single document or multi-document summarization [10,11]. When a document is condensed into a shorter version, it is called single document summarization, whereas condensing a set of documents into a summary is called multi-document summarization. Therefore, summarization of multiple documents can be considered as an extension of summarization of single document [12]. In multi-document summarization, search space is larger compared to single document summarization, which makes it more challenging for extracting important sentences. In that context, multi-document summarization can be considered as an optimization problem with the objective of producing optimal summary containing informative

* Corresponding author.
E-mail addresses: rashmitaroutray@soauniversity.ac.in (R. Rautray), rakesh@iiit-bh.ac.in (R.C. Balabantaray).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.aci.2017.05.003
2210-8327/

sentences of the original documents. Nature inspired optimization based approaches are the suitable choices to address this optimization problem. In literature several meta heuristic techniques such as particle swarm optimization (PSO), differential evolution (DE), harmony search (HS), Cuckoo search (CS) and genetic algorithm (GA) are applied in single or multi-document summarization.

Being inspired by the application of Cuckoo search in other optimization problems [13–34], in this study a novel Cuckoo search algorithm based summarizer is presented for multi-document summarization. Though single document using Cuckoo search algorithm is present in literature [35] but, multi-document summarizer using Cuckoo search is new to this area. Further the model is also compared with Particle Swarm Optimization based summarizer and Cat Swarm Optimization based summarizer. The performance of such models are analyzed over DUC datasets with respect to few summary evaluation metrics such as ROUGE score, inter sentence similarity and readability metric. These evaluation metrics are considered to validate the non-redundancy, cohesiveness and readability of the generated summary.

The structure of paper is organized as follows. Section 2 briefly describes the related works on text summarization problem using global optimization techniques. Section 3 introduces the proposed extractive summarization model. Section 4 presents Cuckoo search based summarizer for solving summarization problem. Next, Section 5 details the numeric calculation for objective function, Section 6 elaborates on experiments and result analysis and finally Section 7 addresses the conclusions.

2. Related works

In this section, a theoretical study of evolutionary algorithms based text summarization and various applications of Cuckoo search algorithm is discussed.

In multi-document summarization, compression of multiple documents, speed of sentence extraction, redundancy between sentences and sentence selection are the critical issues in the formation of useful summaries. In the past, such issues are resolved by statistical tools. But, due to significantly poor performance of statistical tools in text extraction, from 2000 onwards a number of global optimization techniques such as particle swarm optimization (PSO) [2,11,36–38], differential evolution (DE) [1,7,11,12,36,37,39–44], and genetic algorithm (GA) [10,45–51] are proposed by several researchers for improving the performance of sentence selection in document summarization. Initially, the optimization algorithm GA was first used in test summarization problem [45] to retrieve relevant document based on query and relevant judgments. Thereafter in [46], the author has evaluates the efficiency of GA with fitness functions for relevance feedback in information retrieval problem for maintaining the document order. Later on GA based programming technique is used for fuzzy retrieval system to extract information based on query by applying off-line adaptive process [48] and in [49], the author has used GA for text summarization based on sentence score. Each sentence score is obtained through the comparison of each sentence with all other sentences as well as with the document title by cosine measure. The informative features weights are calculated using GA to influence the words relevancy. Word relevancy defines relevancy and rank of the sentences having highest score with respect to a threshold, are selected as summary sentences. A single document generic summary has been extracted based on different sentence features using GA by comparing with some other techniques and were evaluated using ROUGE score [10]. Kogilavani et al. [50] Presents a feature based multi-document generic summarization using GA & clustering to enhance the summary quality by maximizing length, coverage and informativeness while minimizing the redundancy. Whereas, genetic algorithm based document summarization has been proposed to generate optimal summary by combining article sentences and query sentence to achieve satisfied length, high coverage, high informativeness and low redundancy in summary [51,52]. However the GA is providing better result for text summarization. But GA suffers from issues of more parameter tunning [39]. To obtain better summary with less parameter tunning, the authors of [1,7,40,41] have used DE for text summarization problem. Aliguliyev [1] presents a generic document summarizer based on sentence clustering using DE. Whereas in [42], a single document summarizer focuses on sentence feature as key ingredient instead of clustering to extract summary. A summarizer for single document based on clustering has been presented and made comparison of discrete DE and conventional DE for summarization and showed comparison result by the authors of [36]. Alguliev et al. [43] have used DE algorithm to enhance sentence feature based summary by maximizing content coverage, readability and cohesion to improve text readability and informativeness of summary. As the problem of summarization is considered as discrete optimization problem in [43], to solve such problem the author has used adaptive DE to maximize informativeness of summary while reducing the redundancy of summary. In contrast, the summarization problem is considered as p-median problem and Quadratic Boolean programming problem by the authors of [7,40], for that a new variation of DE with self adaptive mutation and crossover parameters and binary DE is used. Where as in [43], adaptive crossover parameter is used for optimizing the summary result. The models discussed in [7,12,39] not only express sentence-to-sentence relationship, but also express summary-to-document and summary-to-subtopics relationships. In all the above cases, DE based summarizer is showing significantly better result than GA based summarizer both for single and multi-document summarization.

Rautray and Balabantaray [37] presents a generic summarizer for single document using particle swarm optimization algorithm, by considering content coverage and redundancy feature as key aspects of summary. For solving such problem, the objective function is designed by taking weighted average of content coverage and redundancy features. Another PSO based single document summarizer is also proposed in [11], which has used the same objective function as described in [37], but by taking features of text as an input arguments instead of sentence weights as input arguments to the model. Binwahlan et al. [2] have presented a PSO based extractive summarizer where expression of ROUGE is used as fitness functions for extraction of summary sentences. The summary based on PSO is also presented by Asgari et al. [38] considering summary features such as content coverage, readability and length. A multi-document summarization system using PSO has been presented in [36] based on the concept of clustering of sentences by calculating inter sentence similarity between sentences and sentence to document set to achieve content coverage and diversity of summary. In contrast, similarity metric also used by Alguliev et al. [44] to achieve content coverage, diversity and length of summary for multiple document sets. Rautray et al. [53] presents cat swarm optimization (CSO) algorithm based multi document summarizer, which takes content coverage, readability and cohesion as key aspects of summary. The summary is evaluated over DUC dataset and compared with two other optimization algorithms such as particle swarm optimization and harmony search algorithm, in which CSO shows competitively better result than other two algorithms.

Cobos et al. [15] have implemented Cuckoo search algorithm for web document clustering or web clustering engine. Cuckoo search uses Balanced Bayesian Information Criteria for fitness function and compared against existing clustering algorithms for web document, Suffix Tree Clustering, Lingo and Bisecting K-mean algorithm. The CS algorithm shows a significant improvement result than other algorithms. A new biodiesel engine have developed by Wong et al. [16] to achieve fewer emissions, low fuel cost and wide operating range of engine using Cuckoo search algorithm. The CS algorithm is compared with PSO algorithm and the result shows that CS is similar to PSO bur with less user defend parameters.

For minimization of power loss and maximization of voltage magnitude, reconfiguration network methodologies using CS algorithm have proposed by Nguyen and Truong [17]. The radial topology of network is maintained by CS algorithm, which is compared with PSO and other compared methods in literature and the result of CS is more noticeable. A combinatorial optimization approach using Cuckoo search algorithm [18] have introduced to minimize possible number of test cases by considering the combination of inputs for detecting defects. Here Cuckoo search algorithm is used to create optimized combinatorial test set. Along with these engineering applications, many other recent applications of Cuckoo Search algorithm are listed in Table 1.

Though various optimization algorithms were proposed in past, but application of Cuckoo search algorithm for developing summarizer is very few in the area of text summarization. Mirshojaei and Masoomi [35] has already addressed summarization problem using Cuckoo search algorithm. But it is applied only for single document summarization. Here, summarization result of Cuckoo search algorithm is compared with the summarization result of particle swarm optimization algorithm, bacterial foraging optimization algorithm and word summarizer in terms of F-score. Among all cases, the F-score of Cuckoo search algorithm is showing comparatively better than the other results.

To the best of the authors' knowledge, no study is available in the open literature with the application of Cuckoo search algorithm for multi-document summarization problem.

3. Multi-document summarization

Multi-document summarization is an automatic process to create a concise and comprehensive document, called summary from multiple documents. The entire procedure of multi-document summarization is divided into three steps such as preprocessing, input representation and summary representation. The overview of summarization system is shown in Fig. 1. Input to the summarization system is multiple documents such as D_1, D_2, ..., D_N. The documents are initially preprocessed, and the result is gone through input representation and summary representation to extract final summary. The detail of summarization process is discussed in the following subsections.

Table 1
Applications of Cuckoo search algorithm in recent years.

Author(s)	Application
Dash et al. [13]	Thermal system
Udayraj et al. [14]	Heat transfer problems
Cobos et al. [15]	Clustering
Ljouad et al. [19]	Object tracker & Kalman filter
Araghi et al. [20]	Traffic signal controller
Wong et al. [16]	Engine optimization
Nguyen et al. [21]	Hydrothermal scheduling
Dash et al. [22]	Thermal system
Nguyen et al. [17]	Network configuration
Abd-Elaziz et al. [23]	Power system
Zineddine [24]	Computer security
Nguyen et al. [25]	Hydrothermal scheduling
Dos Santos et al. [26]	Energy conservation
Wang et al. [27]	Solar radiation
Elkeran [28]	Sheet nesting problem
Bhargava et al. [29]	Phase equilibrium problem
Fateen et al. [30]	Phase stability calculation
Ding et al. [31]	Fuzzy system
Ahmed et al. [18]	Software engineering
Panda et al. [32]	Multilevel thresholding
Bhandari et al. [33]	Satellite image segmentation
Kumar et al. [34]	FIR differentiator design

3.1. Preprocessing

Preprocessing goes through four sub processes.

- **Sentence segmentation:** From the set of input text documents, each individual document D is segmented separately as $D = \{S_1, S_2, \ldots, S_n\}$, where S_j denotes jth sentence in the document for easy extraction of summary sentence, and n is the number of sentences in document.
- **Tokenization:** Terms of each sentence are tokenized as $T = \{t_1, t_2, \ldots, t_m\}$, where t_k for k = 1,2, ..., m. represents all the distinct terms occurring in D and m is the number of terms.
- **Stop word removal:** Most commonly used words in English language such as 'a', 'an', and 'the' which has less important significance with respect to the document are removed.
- **Stemming:** It is a process of chopping off the ends of words to a common base form.

3.2. Input representation

In this section the preprocessed data presented in word form is used to calculate weight (sum of term frequencies) for each sentence known as sentence informative score. The sentence informative score, represented as weight of sentence is further entered as input to the optimization algorithm for implementation. The details of input representation is discussed in Fig. 3.

3.3. Summary representation

The objective of summary representation is generating summary of document sets containing useful information. Through the optimal sentence selection process, the important sentences representing summary is selected by comparing the sentence informative score obtained through optimization algorithm with respect to a pre specified threshold value (see Fig. 4).

4. Cuckoo search based multi-document summarizer

Cuckoo search (CS) is one of latest meta heuristic algorithm, inspired by the species of bird called the Cuckoo. Cuckoos are fascinating birds because of their aggressive reproduction strategy and beautiful sounds, they can make [54–56]. The mature Cuckoos lay their eggs in the nests of other host birds or species [57]. The nest containing each egg represents a solution, and each Cuckoo can lay only one egg that represents new and potentially better solution. The standard Cuckoo search algorithm can be described by three idealized rules: 1) One egg is laid by each Cuckoo in a random nest represents a solution sets; 2) The best eggs contained in the nests will carry over to the next generation; 3) The number of available nests is fixed, and a host bird can discovered an alien egg with a probability (P_a). If this condition satisfies, either the egg can be discarded or abandon the nest by the host, and built a new nest elsewhere.

For implementation point of view, CS algorithm can use the simplest form where each nest has only a single egg. In this case there is no distinction between egg, nest or Cuckoo, as each nest corresponds to one egg which also represents one Cuckoo. The algorithm can be extended to more complicated cases in which each nest has multiple eggs representing a set of solutions.

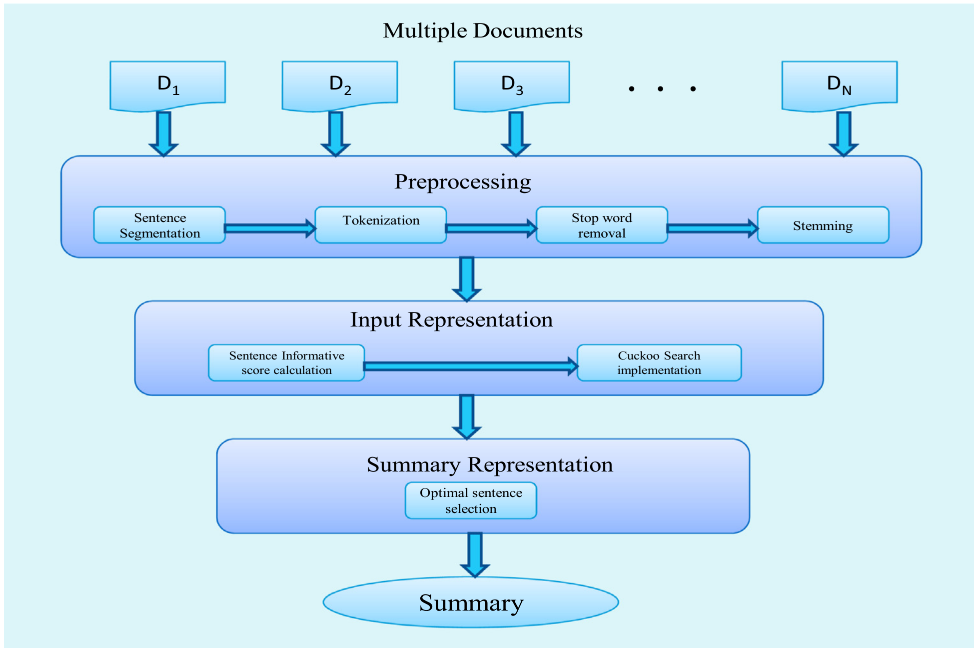

Fig. 1. Overview of summarization system.

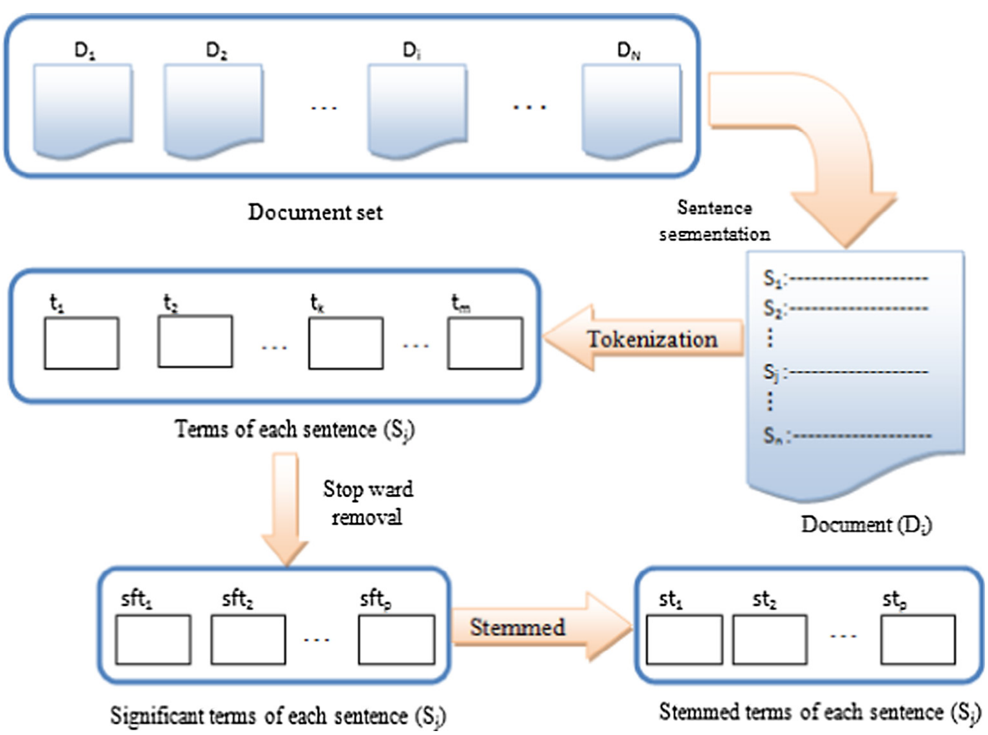

Fig. 2. Flowchart of document preprocessing.

When generating new solutions x_i^{t+1}, a balanced combination of a local random walk and the global explorative random walk is used. This can be controlled by a switching parameter P_a. The local random walk can be written as:

$$x_i^{t+1} = x_i^t + \alpha \times S \otimes H(P_a - \varepsilon) \otimes \left(x_j^t - x_k^t\right) \tag{1}$$

where x_j^t and x_k^t are two different solutions selected randomly by random permutation, $H(u)$ is a Heaviside function, ε is a random number drawn from a uniform distribution and s is the step size. On the other hand, the global random walk is carried out by using Lévy flights. A Lévy flight contains successive random steps [56,58,59], and is characterized by a sequence of rapid jumps, can be represented by the following equation:

$$x_i^{t+1} = x_i^t + \alpha \otimes \text{Lévy}(\lambda) \tag{2}$$

where α is step size, which should be proportional to scale of optimization problem (i.e. $\alpha > 0$), $\otimes$ is entry wise move during multiplication and Lévy(λ) is random numbers drawn from Lévy distribution.

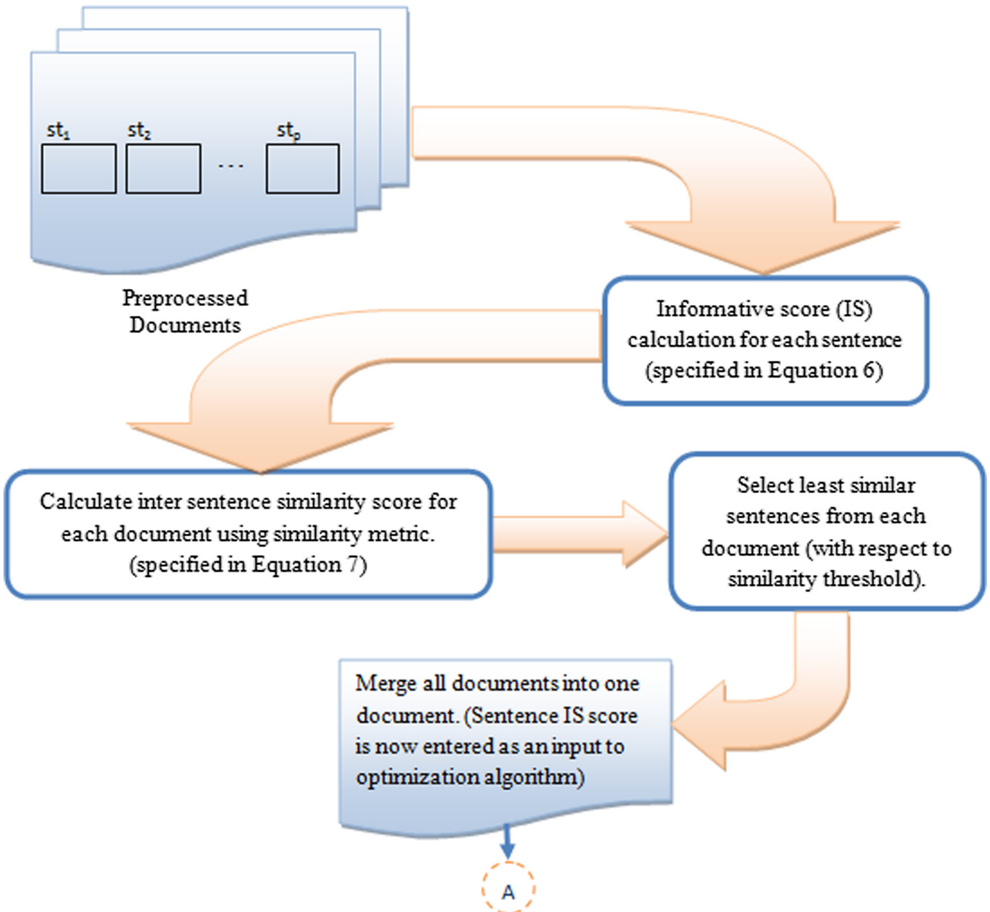

Fig. 3. Flowchart of input representation.

In-addition, the steps involved in MDSCSA is discussed below.

Step 1: Collect a set of multiple documents M, where $M = \{D_1, D_2, \ldots, D_N\}$. *Each* D_i *represents individual document of set M. Length of each* D_i *is represented in terms of number of sentences, which vary from document to document.*
Step 2: Preprocess each text document D_i *using the sentence segmentation, tokenization, stop word removal and stemming steps as shown in* Fig. 2.
Step 3: Calculate the Informative score IS_{jk} *(i.e. the sentence weight derived from the sum of term frequencies) for each sentence* S_j *of the preprocessed document* D_i *using Eq. (3).*

$$IS_{jk} = tf_{jk} \times \log(n/n_k) \tag{3}$$

where IS_{jk} *represents informative score for each sentence* S_j *with respect to term* t_k. tf_{jk} *is the term frequency (i.e. number of times the term* t_k *occurred in sentence* S_j, n_k *denotes the number of sentences in which* t_k *appears. The term* $\log(n/n_k)$ *is referred as inverse sentence frequency used in vector space model for sentence retrieval.*
Step 4: Calculate inter sentence similarity for the preprocessed document D_i *using Eq. (4).*

$$sim(s_i, s_j) = \frac{\sum_{k=1}^{m} IS_{ik} IS_{jk}}{\sqrt{\sum_{k=1}^{m} IS_{ik}^2 \cdot \sum_{k=1}^{m} IS_{jk}^2}}, \quad i,j = 1, \ldots, n \tag{4}$$

Step 5: Select least similar sentences for each D_i *based on a threshold similarity value.*
Step 6: Merge the selected all least similar sentences of each D_i *as a single document* D_{input}.
Step 7: Initialize CS parameters such as population size, rate of alien eggs, step factor (S_f) *and levy exponent* (λ).
Step 8: Use sentence IS score as nest information of each Cuckoo *within the specified search space. Each nest corresponds to a potential solution to the given optimization problem.*
Step 9: Compute the fitness function f_i *for each of these nests as per the given problem using Eq. (3).*
Step 10: The new population of nests is obtained using Lévy flight as specified in Eq. (2).
Step 11: Calculate the fitness f_j *corresponding to the new nests and compare with the fitness* f_i *of the previous nests.*
Step 12: If f_j *is better than* f_i.
Replace the previous nest solution by new nest solution.
Step 13: In the new population, select a fraction P_a *of worst performing nests. Replace these nests by randomly generated ones within the specified search space & build new ones.*
Step14: Compute the fitness function for the new nests obtained.
Step15. Based on the fitness values, record the best performing nests in the current population set. Which are then compared with the best nest obtained until current generation, and replace current best by previous best nest.
Step 16: If the termination criterion is not met, go to Step 9.
Step 17: Select sentences chronologically from the document based on their threshold.

5. Summary evaluation criteria

The objective of the TS problem is to maximize informativeness while reducing redundancy and preserving readability of the

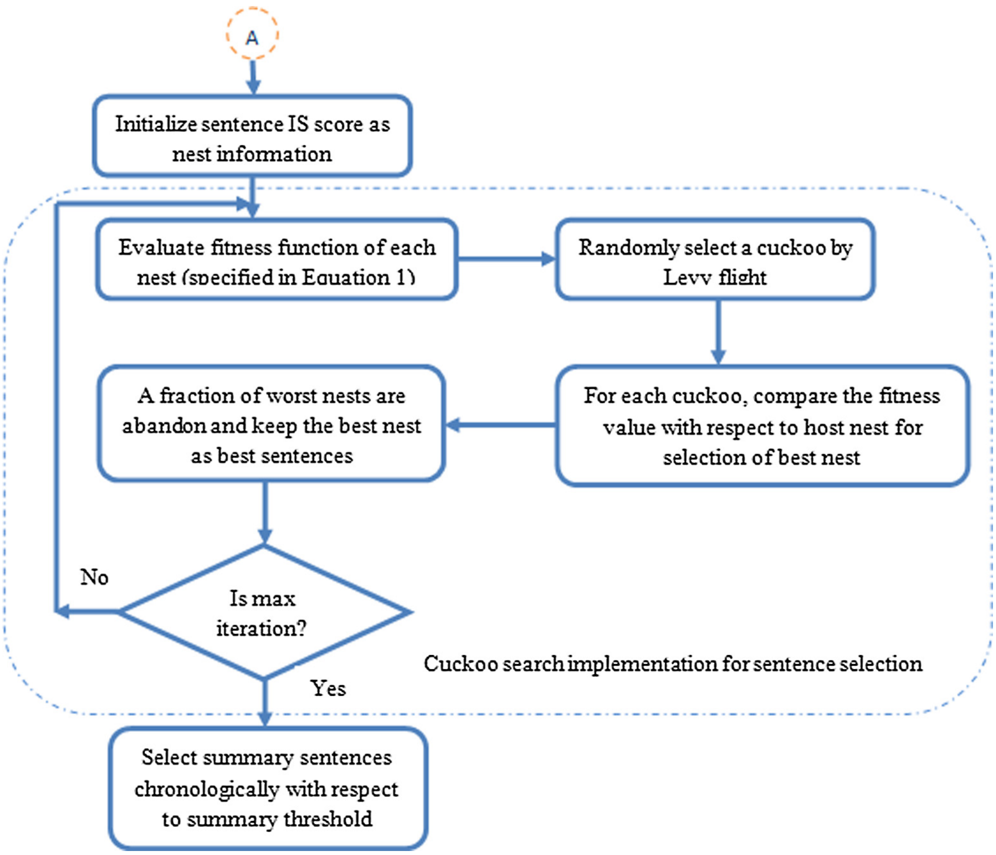

Fig. 4. Flowchart of summary representation.

generated summary. Therefore in this paper, authors have tried to build summaries from document sets with multiple objectives such as content coverage as well as non-redundancy, cohesion and readability, which are explained in following objective function $f(S)$ and that formalize as three sub-functions such as $f_{cov}(S)$, $f_{coh}(S)$ and $f_{read}(S)$ to optimize summary.

$$f(S) = f_{cov}(S) + f_{coh}(S) + f_{read}(S) \tag{5}$$

The objective function (i.e., Eq. (1)) balances the content coverage, cohesion and readability of the summary. The first term evaluates content coverage of the summary. A summary contains a set of relevant sentences, which covers the main content of document set. The main content of document is reflected by the highest weighted sentence or center of the each document. Therefore the content coverage of summary is represented as:

$$f_{cov}(S) = Sim(s_i, O) \quad \mathrm{i} = 1, 2 \ldots \mathrm{n} \tag{6}$$

where O = the center of the main content collection of sentences i.e., $O = \{O_1, O_2, \ldots, O_n\}$ of document sets and O_i is weighted average of sentences of each document. Similarity between S_i and O (specified in Eq. (4)) is evaluated to measure importance of the sentences. Higher similarity values correspond to high content coverage.

The cohesion between the sentences in the summary is connection of ideas both at the sentence level and at the paragraph level. This helps in understanding the complete text in a better way. The ideas of summary select a subset of $s \subset D$ or sentence to sentence relationship that chosen from D. This can be represented as:

$$f_{coh}(S) = 1 - Sim(s_i, s_j) \quad \mathrm{i} \neq \mathrm{j} = 1, 2, \ldots, \mathrm{n} \tag{7}$$

The higher value of $f_{coh}(S)$ specifies high connection between sentences and vice versa.

The summary readability select a subset of $s \subset D$ that maximizes the inter sentence relationship of s chosen from D. As $f_{read}(S)$ *measures* similarity (specified in Eq. (4)) between S_i *and* S_j, the higher value of $f_{read}(S)$ specifies higher readability of the summary, which is defined as:

$$f_{read}(S) = Sim(s_i, s_j) \quad \mathrm{i} \neq \mathrm{j} = 1, 2, \ldots, \mathrm{n} \tag{8}$$

6. Experiment and result analysis

This section conduct experiments to test proposed summarization system empirically. The MDSCSA is compared with CSOS and PSOS multi-document summarizer with respect to two years of DUC datasets. All the summarizer models are implemented in MATLAB Version 2014a) in a system with Window 7 operating system. After obtaining the simulation result, the analysis of summary result has been carried out using ROUGE tool in terms of ROUGE score.

6.1. Dataset

The open bench mark datasets from DUC (Document Understanding Conference) are used for the evaluation of text extraction result. Table 2 provides a short description of DUC data sets. By the step of data preprocessing, less significant words or stop words from the original documents are removed by comparing with the available stop word list in net and the terms are stemmed using the most common stemmer in English called Porter's stemmer.

6.2. Controlling parameters

Controlling parameters of any optimization algorithm are application oriented. Thus, there is no fixed value is assign to these

Table 2
Dataset description.

Data set parameters	Size (DUC2006)	Size (DUC2007)
Number of clusters	50	45
Number of documents in each clusters	25	25
Average no. of sent. per doc	30.12	37.5
Maximum no. of sent. per doc	79	125
Minimum no. of sent. per doc	5	9
Data source	AQUAINT	AQUAINT
Summary length (in words)	250	250

parameters. Therefore derivation of parameters is obtained through number of simulations. For this text extraction problem, the controlling parameters of MDSCSA, CSOS and PSOS models are present in Table 3.

6.3. Evaluation metric

For summary evaluation, ROUGE-1.5.5 package developed by [60] is used in this study. It is used as the evaluation metric for text summarization. ROUGE includes different methods such as ROUGE-L, ROUGE-N, ROUGE-S, ROUGE-W and ROUGE-SU to measure the n-gram match between systems generated summaries and human summaries. Here ROUGE-N metric compares N-grams of two summaries, and counts the number of matches:

$$ROUGE-N = \frac{\sum_{S \in Summ_{ref}} \sum_{N-gram \in S} Count_{match}(N-gram)}{\sum_{S \in Summ_{ref}} \sum_{N-gram \in S} Count(N-gram)} \quad (9)$$

where N stands for the length of the N-gram, count match (N-gram) is the highest number of N-grams co-occurring in candidate summary and reference-summaries. Count (N-gram) is the number of N-grams in the reference summaries.

Furthermore, sensitivity, positive predictive value (PPV) and summary accuracy (Summary_{acc}) are used for summary evaluation. The sensitivity, PPV and Summary_{acc} of summary are evaluated based on the outcomes of candidate summary (Candidate_{sum}), reference summary (Reference_{sum}), true sentences (True_{sen}) and least significant sentences (LS_{sen}). The summary which is generated by our proposed summarizer is called candidate summary. Whereas, the summary is refer for an evaluation, called reference summary. In both the summary, the common sentences are referred as true sentences. But the sentences, neither in Candidate_{sum} nor in Reference_{sum} is called LS_{sen}. Sensitivity, PPV and Summary_{acc} are calculated using the following equations.

$$Sensitivity = \frac{|True_{sen}|}{|True_{sen}| + |Reference_{sum}|} \quad (10)$$

$$PPV = \frac{|True_{sen}|}{|True_{sen}| + |Candidate_{sum}|} \quad (11)$$

$$S_{acc} = \frac{|True_{sen}| + |LS_{sen}|}{|True_{sen}| + |LS_{sen}| + |Reference_{sum}| + |Candidate_{sum}|} \quad (12)$$

6.4. Performance analysis

This section analyses the performance of various models on the basis of three summary evaluation criteria as discussed in Section 4.

6.4.1. Observation 1 (based on ROUGE-N)

The summary performance has been evaluated by using ROUGE-N with two *N* values such as ROUGE-1 and ROUGE-2 metrics. These matrices are highly correlated with the human judgments. ROUGE-1 measures the overlap of unigrams between the system summary and the manual summaries created by human while ROUGE-2 compares the overlap of bigrams [43]. The ROUGE-N evaluation is done based on content coverage, cohesiveness and text readability of summary. A model providing higher ROUGE metric indicates higher similarity of the generated summary with respect to the original document sets. Though the ROUGE-N value is represented in terms of three different metrics such as precision, recall and F-measure value, F-measure is assumed to have more significance for selection of a summary. In this study the model selection is done based on the best F-measure of the ROUGE-N values. Table 4 shows the statistical analysis in term of worst, mean and best of F-measure of ROUGE-1 and ROUGE-2 evaluation metrics observed for the PSOS, CSOS and MDSCSA algorithm on DUC 2006 and DUC 2007 document set respectively. The evaluation metrics are observed for the system generated summaries (summary generated by PSOS, CSOS and MDSCSA) with human generated summaries present in DUC. From the comparison of F measure it is observed that the best F measure value with respect to ROUGE-1 for all the three optimization algorithms are falling within the range 0.41–0.44 and with respect to ROUGE-2 it is within the range 0.07–0.13 for DUC 2006 dataset. Similarly for DUC 2007 dataset, the best F measure value with respect to ROUGE-1 is falling within the range 0.40–0.43 and with respect to ROUGE-2 it is within the range 0.08–0.10. Though the values are data dependent it is clearly observed that, Cuckoo search is providing better F measure values (best of statistical analysis) for both the ROUGE scores on both the datasets. Further the precision, recall and F measure of both the ROUGE scores with respect to two datasets is specified in Table 5.

Analyzing the three matrices of ROUGE-N score and document classification metrics, it is clearly observed that MDSCSA is providing better result compared to PSOS and CSOS with respect to ROUGE-1 and ROUGE-2 for both the datasets. The F measure value of ROUGE-N score is dependent on both recall and precision value. Similarly summary accuracy is dependent on both sensitivity and PPV score. So instead of evaluating the summarizers with respect to precision, recall, sensitivity and PPV score separately, the model validation is done based on the F measure value and summary accuracy value (see Table 6).

6.4.2. Observation 2 (based on cohesion)

Cohesion is an essential element for the reader to be clear and to achieve its final purpose. It refers to the degree to which sentences (or even different parts of one sentence) are connected so

Table 3
Parameters used for PSO, CSO and CS based summarizer.

PSOS		CSOS		MDSCSA	
Population size	50 docs	Population size	50 docs	Population size	50 docs
C1	[0,2]	SMP	3	Rate of alien eggs (P_a)	0.75
C2	[0,2]	CDC	0.2	Step size (S_f)	0.5
*V*min, *V*max	[0,1]	SRD	0.2	Levy exponent (λ)	0.8
W	0.45	Mixture ratio (MR)	0.5		
		w, C	0.5, 4		

Table 4
Performance comparisons of PSOS, CSOS and MDSCSA summarizer based on ROUGE-N (F measure) metric for DUC2006 and DUC2007 data.

Dataset	Evaluation metric	Optimization algorithm	Worst	Mean	Best
DUC 2006	Rouge-1	PSOS	0.39087	0.4009	0.41127
		CSOS	0.4003	0.4070	0.4229
		MDSCSA	0.40422	0.4115	0.4311
	Rouge-2	PSOS	0.05848	0.0651	0.0784
		CSOS	0.0714	0.0831	0.09033
		MDSCSA	0.07677	0.0864	0.13986
DUC 2007	Rouge-1	PSOS	0.3916	0.3991	0.40967
		CSOS	0.3908	0.4098	0.4207
		MDSCSA	0.4000	0.4116	0.4243
	Rouge-2	PSOS	0.0743	0.0758	0.0762
		CSOS	0.0809	0.0881	0.08903
		MDSCSA	0.0817	0.0892	0.1034

Table 5
Precision, recall and F measure of ROUGE-N score for both the dataset.

Dataset	Evaluation metric	Optimization algorithm	Recall	Precision	F measure
DUC 2006	Rouge-1	PSOS	0.44151	0.38491	0.41127
		CSOS	0.43098	0.41520	0.4229
		MDSCSA	0.43655	0.4258	0.4311
	Rouge-2	PSOS	0.08255	0.07469	0.0784
		CSOS	0.0995	0.08271	0.09033
		MDSCSA	0.12346	0.16129	0.13986
DUC 2007	Rouge-1	PSOS	0.44679	0.37825	0.40967
		CSOS	0.46158	0.38662	0.4207
		MDSCSA	0.4583	0.3951	0.4243
	Rouge-2	PSOS	0.0841	0.0697	0.0762
		CSOS	0.0924	0.0859	0.08903
		MDSCSA	0.1093	0.09824	0.1034

Table 6
Performance comparison of PSO, CSO and CS summarizer based on sensitivity, PPV and summary accuracy for both the dataset.

Dataset	Optimization algorithm	Evaluation metrics		
		Sensitivity	PPV	$Summary_{acc}$
DUC 2006	PSOS	0.5	0.4	0.9734
	CSOS	0.56	0.5294	0.9800
	MDSCSA	0.6	0.5708	0.99
DUC 2007	PSOS	0.5	0.3529	0.9808
	CSOS	0.5833	0.5	0.9904
	MDSCSA	0.62	0.54	0.9951

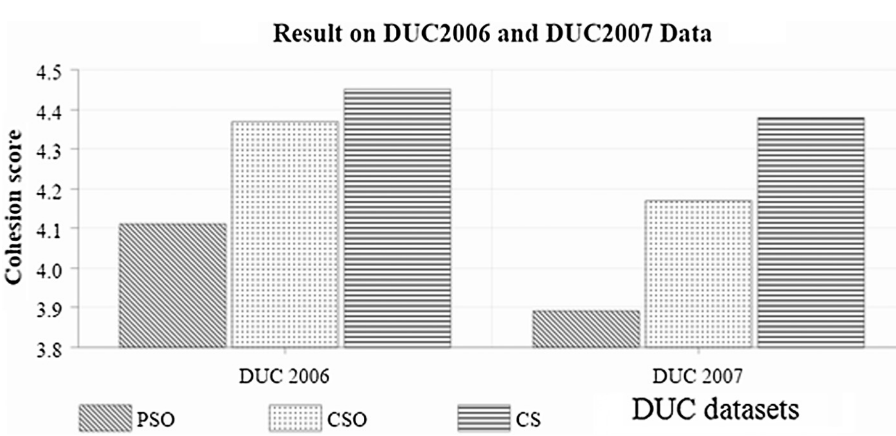

Fig. 5. Cohesion score comparison on DUC2006 and DUC2007 dataset.

that the flow of ideas is easy to follow. Cohesiveness in summary does not mean just "grammatically correctness" of sentences but cohesive summary refers to the connection of ideas both at the sentence level and at the paragraph level. Therefore cohesion of consecutive sentences helps in understanding the complete text better [40]. The common metric used to compute cohesiveness of summary is cosine similarity by considering average similarity of the sentences. Fig. 5 shows cohesion score of different methods

Table 7
Readability metric formulas.

Readability metric	Formula	Equation no.
Flesch Kincaid Grade Level (FKGL)	0.39 × (words/sentences) + 11.8 × (syllables/words) – 15.59	(13)
Gunning fog score (FOG)	0.4 (Average Sentence Length + Percentage of Hard Words)	(14)
SMOG Index (SMOG)	1.0430 × sqrt(30 × complex words/sentences) + 3.1291	(15)
Coleman Liau (CL)	5.89 × (characters/words) – 0.3 × (sentences/words) – 15.8	(16)
Automated readability index (ARI)	4.71 × (characters/words) + 0.5 × (words/sentences) – 21.43	(17)

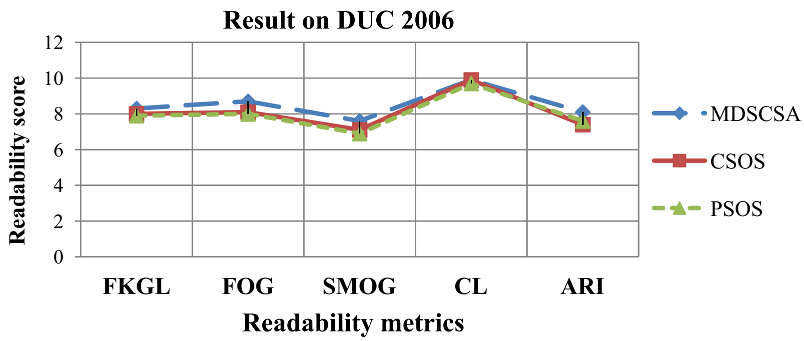

Fig. 6. Readability score of different methods on DUC2006 dataset.

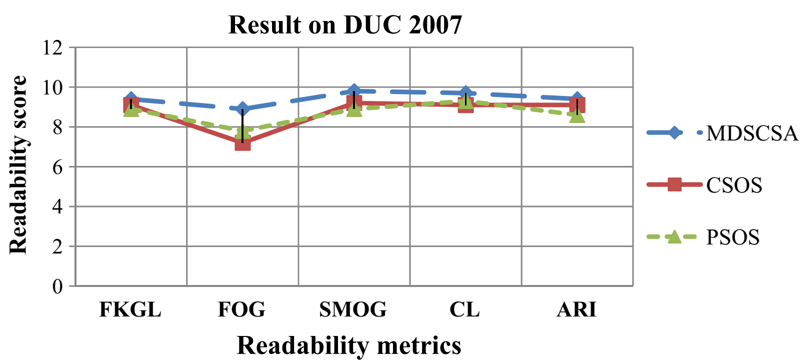

Fig. 7. Readability score of different methods on DUC2007 dataset.

on DUC datasets. From the analysis, it is observed that MDSCSA has shown comparatively better cohesion value than the PSOS and CSOS for both the datasets.

6.4.3. Observation 3 (based on readability)

This experiment involves readability of summary, which means "how easily materials can be read and understood? This depends on several factors including the average length of sentences, the number of new words contained, and the grammatical complexity of the language used in a passage" [61]. Readability can be calculated by the formula discussed in Table 7. Readability is estimated in terms of the number of years of education one needs to have to comprehend that text [62]. The higher value of readability metric supports easy reading and understanding of generated summary whereas lower value creates difficulty in reading and understanding of the summary. The readability score of three different summarizers for DUC 2006 dataset and DUC 2007 dataset is shown in Figs. 6 and 7 respectively. From the analysis, it is clearly observed that, for DUC 2006 dataset MDSCSA is providing better readability score with respect to FKGL, FOG, SMOG and ARI metrics compared to PSOS and CSOS and for CL metric all the three summarizers are producing almost same result. For DUC 2007 dataset MDSCSA is providing better readability score with respect to all the metrics compared to both PSOS and CSOS summarizer.

7. Conclusion

This paper focuses on a Cuckoo search based multi-document summarizer to create a generic extractive summary. The summarizer is also compared with particle swarm optimization based summarizer and cat swarm optimization based summarizer. The performance of all discussed summarizers are evaluated in terms of ROUGE score, inter sentence similarity and readability metric to validate non-redundancy, cohesiveness and readability of the summary respectively on a benchmark dataset called as Document Understanding Conference datasets in three experiments. Observation 1 and 2 discusses non-redundancy and cohesiveness of summary, where in most of the cases Cuckoo search based model is showing better ROUGE score. Similarly in readability test discussed in observation 3, MDSCSA is also showing better readable score of the summary in Figs. 6 and 7 compared to PSOS & CSOS based model. From the above observations, it can be concluded that the performance of MDSCSA is significantly better than the CSOS and PSOS algorithm in summary generation.

Controlling of evolutionary algorithm parameters are purely data dependent in the experiment of any application. As Cuckoo search algorithm is an evolutionary approach, thus the limitation of this approach is its controlling parameters. Therefore more systematic approach of parameter setting will be explored in our future work. The performance of this approach can also be examined using other competent nature inspired algorithms.

References

[1] R.M. Aliguliyev, A new sentence similarity measure and sentence based extractive technique for automatic text summarization, Expert Syst. Appl. 36 (4) (2009) 7764–7772.

[2] M.S. Binwahlan, N. Salim, L. Suanmali, Swarm based text summarization, in: Computer Science and Information Technology-Spring Conference, 2009, IACSITSC'09, International Association of, IEEE, 2009, April, pp. 145–150.

[3] K. Ježek, J. Steinberger, Automatic text summarization (the state of the art 2007 and new challenges), in: Proceedings of Znalosti, 2008, February, pp. 1–12.

[4] E. Lloret, M. Palomar, Text summarisation in progress: a literature review, Artif. Intell. Rev. 37 (1) (2012) 1–41.

[5] M. Mendoza, S. Bonilla, C. Noguera, C. Cobos, E. León, Extractive single-document summarization based on genetic operators and guided local search, Expert Syst. Appl. 41 (9) (2014) 4158–4169.

[6] H. Oliveira, R. Ferreira, R. Lima, R.D. Lins, F. Freitas, M. Riss, S.K. Simske, Assessing shallow sentence scoring techniques and combinations for single and multi-document summarization, Expert Syst. Appl. 65 (2016) 68–86.

[7] R.M. Alguliev, R.M. Aliguliyev, M.S. Hajirahimova, GenDocSum+ MCLR: Generic document summarization based on maximum coverage and less redundancy, Expert Syst. Appl. 39 (16) (2012) 12460–12473.

[8] I. Mani, M.T. Maybury, Advances in Automatic Text Summarization, MIT Press, 1999.

[9] X. Wan, J. Xiao, Exploiting neighborhood knowledge for single document summarization and keyphrase extraction, ACM Trans. Inf. Syst. (TOIS) 28 (2) (2010) 8.

[10] M.A. Fattah, F. Ren, GA, MR, FFNN, PNN and GMM based models for automatic text summarization, Comput. Speech Lang. 23 (1) (2009) 126–144.

[11] R. Rautray, R.C. Balabantaray, A. Bhardwaj, Document summarization using sentence features, Int. J. Inf. Retrieval Res. (IJIRR) 5 (1) (2015) 36–47.

[12] R.M. Alguliev, R.M. Aliguliyev, N.R. Isazade, CDDS: constraint-driven document summarization models, Expert Syst. Appl. 40 (2) (2013) 458–465.

[13] P. Dash, L.C. Saikia, N. Sinha, Comparison of performances of several Cuckoo search algorithm based 2DOF controllers in AGC of multi-area thermal system, Int. J. Electr. Power Energy Syst. 55 (2014) 429–436.

[14] Udayraj, K. Mulani, P. Talukdar, A. Das, R. Alagirusamy, Performance analysis and feasibility study of ant colony optimization, particle swarm optimization and cuckoo search algorithms for inverse heat transfer problems, Int. J. Heat Mass Transf. 89 (2015) 359–378.

[15] C. Cobos, H. Muñoz-Collazos, R. Urbano-Muñoz, M. Mendoza, E. León, E. Herrera-Viedma, Clustering of web search results based on the cuckoo search algorithm and Balanced Bayesian Information Criterion, Inf. Sci. 281 (2014) 248–264.

[16] P.K. Wong, K.I. Wong, C.M. Vong, C.S. Cheung, Modeling and optimization of biodiesel engine performance using kernel-based extreme learning machine and cuckoo search, Renewable Energy 74 (2015) 640–647.

[17] T.T. Nguyen, A.V. Truong, Distribution network reconfiguration for power loss minimization and voltage profile improvement using cuckoo search algorithm, Int. J. Electr. Power Energy Syst. 68 (2015) 233–242.

[18] B.S. Ahmed, T.S. Abdulsamad, M.Y. Potrus, Achievement of minimized combinatorial test suite for configuration-aware software functional testing using the Cuckoo search algorithm, Inf. Softw. Technol. (2015).

[19] T. Ljouad, A. Amine, M. Rziza, A hybrid mobile object tracker based on the modified Cuckoo Search algorithm and the Kalman Filter, Pattern Recogn. 47 (11) (2014) 3597–3613.

[20] S. Araghi, A. Khosravi, D. Creighton, Intelligent cuckoo search optimized traffic signal controllers for multi-intersection network, Expert Syst. Appl. 42 (9) (2015) 4422–4431.

[21] T.T. Nguyen, D.N. Vo, Modified cuckoo search algorithm for short-term hydrothermal scheduling, Int. J. Electr. Power Energy Syst. 65 (2015) 271–281.

[22] P. Dash, L.C. Saikia, N. Sinha, Comparison of performances of several FACTS devices using Cuckoo search algorithm optimized 2DOF controllers in multi-area AGC, Int. J. Electr. Power Energy Syst. 65 (2015) 316–324.

[23] A.Y. Abd-Elaziz, E.S. Ali, Cuckoo search algorithm based load frequency controller design for nonlinear interconnected power system, Int. J. Electric Power Energy Syst., 73 C (2015) 632–643.

[24] M. Zineddine, Vulnerabilities and mitigation techniques toning in the cloud: A cost and vulnerabilities coverage optimization approach using Cuckoo search algorithm with Lévy flights, Comput. Security 48 (2015) 1–18.

[25] T.T. Nguyen, D.N. Vo, A.V. Truong, Cuckoo search algorithm for short-term hydrothermal scheduling, Appl. Energy 132 (2014) 276–287.

[26] L. Dos Santos Coelho, C.E. Klein, S.L. Sabat, V.C. Mariani, Optimal chiller loading for energy conservation using a new differential cuckoo search approach, Energy 75 (2014) 237–243.

[27] J. Wang, H. Jiang, Y. Wu, Y. Dong, Forecasting solar radiation using an optimized hybrid model by Cuckoo Search algorithm, Energy 81 (2015) 627–644.

[28] A. Elkeran, A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering, Eur. J. Oper. Res. 231 (3) (2013) 757–769.

[29] V. Bhargava, S.E.K. Fateen, A. Bonilla-Petriciolet, Cuckoo search: a new nature-inspired optimization method for phase equilibrium calculations, Fluid Phase Equilib. 337 (2013) 191–200.

[30] S.E.K. Fateen, A. Bonilla-Petriciolet, A note on effective phase stability calculations using a Gradient-Based Cuckoo Search algorithm, Fluid Phase Equilib. 375 (2014) 360–366.

[31] X. Ding, Z. Xu, N.J. Cheung, X. Liu, Parameter estimation of Takagi-Sugeno fuzzy system using heterogeneous cuckoo search algorithm, Neurocomputing 151 (2015) 1332–1342.

[32] R. Panda, S. Agrawal, S. Bhuyan, Edge magnitude based multilevel thresholding using Cuckoo search technique, Expert Syst. Appl. 40 (18) (2013) 7617–7628.

[33] A.K. Bhandari, V.K. Singh, A. Kumar, G.K. Singh, Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy, Expert Syst. Appl. 41 (7) (2014) 3538–3560.

[34] M. Kumar, T.K. Rawat, Optimal design of FIR fractional order differentiator using cuckoo search algorithm, Expert Syst. Appl. 42 (7) (2015) 3433–3449.

[35] S.H. Mirshojaei, B. Masoomi, Text summarization using cuckoo search optimization algorithm, J. Comput. Robotics 8(2) (2015) 19–24.

[36] R.M. Alguliev, R.M. Aliguliyev, C.A. Mehdiyev, Sentence selection for generic document summarization using an adaptive differential evolution algorithm, Swarm Evolutionary Comput. 1 (4) (2011) 213–222.

[37] R. Rautray, R.C. Balabantaray, Comparative study of DE and PSO over document summarization, in: Intelligent Computing, Communication and Devices, Springer India, 2015, pp. 371–377.

[38] H. Asgari, B. Masoumi, O.S. Sheijani, Automatic text summarization based on multi-agent particle swarm optimization, in: Intelligent systems (ICIS), 2014 Iranian conference on, IEEE, 2014, February, pp. 1–5.

[39] R.M. Alguliev, R.M. Aliguliyev, N.R. Isazade, DESAMC+ DocSum: Differential evolution with self-adaptive mutation and crossover parameters for multi-document summarization, Knowl.-Based Syst. 36 (2012) 21–38.

[40] A. Abuobieda, N. Salim, M.S. Binwahlan, A.H. Osman, Differential evolution cluster-based text summarization methods, in: Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on, IEEE, 2013, August, pp. 244–248.

[41] S. Karwa, N. Chatterjee, Discrete differential evolution for text summarization, in: Information Technology (ICIT), 2014 International Conference on, IEEE, 2014, December, pp. 129–133.

[42] K. Nandhini, S.R. Balasundaram, Extracting easy to understand summary using differential evolution algorithm, Swarm Evolutionary Comput. 16 (2014) 19–27.

[43] R.M. Alguliev, R.M. Aliguliyev, N.R. Isazade, Multiple documents summarization based on evolutionary optimization algorithm, Expert Syst. Appl. 40 (5) (2013) 1675–1689.

[44] R.M. Alguliev, R.M. Aliguliyev, M.S. Hajirahimova, C.A. Mehdiyev, MCMR: Maximum coverage and minimum redundant text summarization model, Expert Syst. Appl. 38 (12) (2011) 14514–14522.

[45] M.D. Gordon, Probabilistic and genetic algorithms for document retrieval, Commun. ACM 31 (10) (1988) 1208–1218.

[46] C. López-Pujalte, V.P. Guerrero-Bote, F. de Moya-Anegón, Order-based fitness functions for genetic algorithms applied to relevance feedback, J. Am. Soc. Inform. Sci. Technol. 54 (2) (2003) 152–160.

[47] O.C. García, F. de Moya Anegón, C. Zarco, A GA-P algorithm to automatically formulate extended Boolean queries for a fuzzy information retrieval system, Mathware Soft Comput. 7 (2) (2000) 309–322.

[48] R.M. Alguliev, R.M. Aliguliyev, Effective summarization method of text documents, in: Web Intelligence, 2005, Proceedings, The 2005 IEEE/WIC/ACM International Conference on, IEEE, 2005, September, pp. 264–271.

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

[50] Y.X. He, D.X. Liu, D.H. Ji, H. Yang, C. Teng, Msbga: A multi-document summarization system based on genetic algorithm, in: Machine Learning and Cybernetics, 2006 International Conference on, IEEE, 2006, August, pp. 2659–2664.

[51] X. Zhao, J. Tang, Query-focused summarization based on genetic algorithm, in: 2010 International Conference on Measuring Technology and Mechatronics Automation, IEEE, 2010, March, pp. 968–971.

[52] A.E. Eiben, J.E. Smith, Introduction to Evolutionary Computing Eiben, Springer Publisher, 2015.

[53] R. Rautray, R.C. Balabantaray, Cat swarm optimization based evolutionary framework for multi document summarization, Phys. A: Stat. Mech. Appl. 477 (2017) 174–186.

[54] X.S. Yang, S. Deb, Cuckoo search via Levy flights, in: Proceedings of World Congress on Nature and Biologically Inspired Computing (NaBIC 2009), IEEE Publications, USA, 2009, pp. 210–214.

[55] X.S. Yang, S. Deb, Engineering optimisation by cuckoo search, Int. J. Math. Modelling Numer. Optimisation 1 (4) (2010) 330–343.

[56] X.S. Yang, S. Deb, Cuckoo search: recent advances and applications, Neural Comput. Appl. 24 (1) (2014) 169–174.

[57] A.B. Mohamad, A.M. Zain, N.E. Nazira Bazin, Cuckoo search algorithm for optimization problems—a literature review and its applications, Appl. Artificial Intelligence 28 (5) (2014) 419–448.
[58] W.C.E. Lim, G. Kanagaraj, S.G. Ponnambalam, Cuckoo search algorithm for optimization of sequence in pcb holes drilling process, in: Emerging Trends in Science, Engineering and Technology, Springer India, 2012, pp. 207–216.
[59] G. Kanagaraj, S.G. Ponnambalam, N. Jawahar, Reliability-based total cost of ownership approach for supplier selection using cuckoo-inspired hybrid algorithm, Int. J. Adv. Manuf. Technol. (2014) 1–16.
[60] C.Y. Lin, E. Hovy, Automatic evaluation of summaries using n-gram co-occurrence statistics, in: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, Association for Computational Linguistics, 2003, May, pp. 71–78.
[61] M. Zamanian, P. Heydari, Readability of texts: state of the art, Theory Practice Language Stud. 2 (1) (2012) 43.
[62] J. Kondru, Using part of speech structure of text in the prediction of its readability, Doctoral dissertation, University of Texas at Arlington, 2006.